THE SECRET GUIDE TO COMPUTERS

Volume 4: Hassles in Systems

The wise way to build up your computer system...and your career

10th edition

Donated to
Internet Archive by
Kay Savetz

Dear You!

Welcome to volume 4 of The Secret Guide to Computers. Before reading this volume, read volumes 1, 2, and 3— especially volume 3, which explains hardware jargon and the differences between computer manufacturers. To order those volumes, use the coupon on the back page.

If you've been waiting a long time for volume 4, I apologize for the delay; but the delay has actually worked to your advantage. It's enabled me to include comments that are more up-to-date and wiser. I expect this volume will thrill you.

It explains the newest tricks for buying a computer and for enhancing your career. It answers most of your questions. If you have more questions, phone me at 617-266-8128. You can call 24 hours: I'm almost always in, and I sleep only lightly. At your service, your computer butler, Russ Walter

P.S. If you're receiving this book in June or in the beginning of July, and want to become a computer expert quickly, read pages 72-75 and then phone me *immediately*, before you're too late!

TABLE OF CONTENTS

BEFORE YOU BUY

Summary: Radio Shack, Apple, Atari, low-cost computers, larger computers 2
Groundwork: acquisition, minimum requirements, dialects of BASIC 6

POPULAR COMPANIES

Radio Shack: model 3, color, pocket, discounts, nasty, used, imitations 11
Apple: prices, imitations, slots, black Apples, repairs, Apple 3, mistakes 16
Commodore: prices, disk drives, VIC, new address 19

ADVANCED PURCHASING

Which system to buy: low-cost computers, 80-column computers, multi-user 21
Printers: devilish dozen, from Sinclair to Epson to Spinwriter, obituaries 26

PLAY WITH BITS

Number systems: computers aren't human, binary, octal, hexadecimal 32
Character codes: ASCII, EBCDIC, bytes, nibbles 36
Chips: generations, DIPs, prices, bipolar vs. MOS, kinds of RAM & ROM 38

PLAY WITH SYSTEMS

Scheduling the users: sharing the CPU, memory, I/O devices 42
Personality: computers have personalities, boss, conversationalist, drudge 46
Operating systems: IBM, DEC, Radio Shack, Apple, CP/M 47

GROWTH OF A PROFESSION

Jobs: programmer, management, sales, original jobs 49
History: before computers, four generations, boredom, personal computers 56
Spelling: errors, choices 65

WHAT TO READ

Books: on BASIC & systems & applications & languages, where to buy 66
Periodicals: newspapers, monthly micro magazines 71

COURSES TO TAKE

Summer course: best course, many versions, consultants, guest speakers 72
Blitz course: super-intensive weekend, your own computer, tapes, refunds 76

This book is © 1981 by Russ Walter, 92 Saint Botolph St., Boston, MA 02116. You can get permission to reprint it, for your own purposes, if you phone us.

Summary

HOW TO BUY A COMPUTER

First, decide *why* you want a computer. What do you want to use the computer for? Then decide how many people will want to use the computer simultaneously: if the answer is "many", you'll need to buy a computer that has many terminals, or you'll need to hook many single-user computers together, to form a network. Then estimate how large your files are: how many megabytes of disk will they require?

Then hunt for the lowest-cost computer that meets your needs: hunt for a computer that has enough megabytes, allows enough simultaneous users, and comes with adequate software and programs. Decide whether you can afford it: if not, don't computerize yet.

Then examine other computers that are slightly more expensive but slightly better, and decide whether their "extras" are useful enough to be worth paying for. Last but not least, decide whether to buy the computer from your local dealer (at list price) or mail-order (at discount).

Begin by looking at microcomputers. For most readers of this book, microcomputers are adequate. But if your needs are very large— if you need to let many people share many megabytes— you might have to shell out more money, and get a minicomputer or maxicomputer instead.

The most popular microcomputer manufacturers are Radio Shack and Apple. Begin by looking at their products, then consider the competition.

RADIO SHACK

Radio Shack's first computer was called the "model 1". Then Radio Shack invented the "model 2", which was better than the model 1 but much more expensive. Then Radio Shack invented the "model 3", which is a pleasant compromise between the model 1 and the model 2: the model 3 is almost as good as the model 2, yet almost as cheap as the model 1.

Radio Shack has stopped manufacturing the model 1. So if you want to buy a business-oriented computer from Radio Shack, you have two choices: buy the model 3, or buy the model 2. For most readers of this book, the model 3 is adequate; but if you need larger disks or more characters on the screen, you'll have to shell out more money, and get the model 2 instead.

If you're so short of money that you can't afford even the model 3, you have several alternatives. You can try to buy a used model 1. You can buy imitations of the model 1; the imitations are sold by several independent companies, many of whom Radio Shack is suing. You can buy one of Radio Shack's new extra-low-cost computers: the Color Computer (which is cheaper than the model 1) or the Pocket Computer (which is even cheaper!).

The Pocket Computer is manufactured for Radio Shack by Sharp; if you don't want to buy it from Radio Shack, you can buy it from Sharp directly. The Pocket Computer is cute but awkward: for most readers of this book, the Pocket Computer is inadequate.

The Color Computer, on the other hand, is okay, if your main goal is educational: just to learn how computers work and to practice programming. Since the Color Computer doesn't have disks yet, it isn't yet adequate for running a business. The Color Computer lists for $399, but you can get it for $310 from discount dealers. To use it, you must also buy a television and a tape recorder— or use the ones you already have at home.

APPLE

Apple's first computer was called the Apple 1 and is obsolete. Then came the Apple 2, which is obsolescent. Then came the Apple 2-plus, which is the most popular. Then came the Apple 3, which is the fanciest but ridiculously overpriced. So if you want to buy an Apple, your best bet is the Apple 2-plus.

APPLE 2-PLUS VERSUS RADIO SHACK'S MODEL 3

The Apple 2-plus is similar to Radio Shack's model 3, but is slightly more expensive and slightly stupider. (To make the Apple 2-plus act as smart as Radio Shack's model 3, you must insert many circuit cards into the Apple; the circuit cards are extra-cost options, and jack the price up even higher.) But even though The Apple 2-plus gives you less-for-your-money than Radio Shack's model 3, the Apple 2-plus is outselling Radio Shack's model 3, for these reasons....

The Apple has color graphics. (The Radio Shack is black-and-white.)

More programs have been written for the Apple than for Radio Shack's model 3 (because the model 3 is too new).

The Apple is more expandable; you can insert circuit cards into the Apple more easily than into the Radio Shack. (But the circuit cards are very expensive, and Radio Shack's price already includes most of the circuit cards you want.)

Politics and marketing: Apple Computer Company has done a cleverer job of salesmanship than Radio Shack during the past few months.

People used to say, "Apple computers are more reliable than Radio Shack's." That statement is no longer true. In fact, based on the phone calls I've been receiving and the articles I've read, the situation seems to have reversed itself: Radio Shack's new computers (the Pocket Computer, Color Computer, model 3, and model 2) are all *more* reliable than the Apple 2-plus. However, the Apple 2-plus is still reliable enough for most purposes.

Here's a summary of the Apple versus Radio Shack controversy....

In favor of Radio Shack: Radio Shack's model 3 costs less than the Apple 2-plus; Radio Shack includes more equipment and intelligence in its price than Apple; Radio Shack is probably adequate for your needs.

In favor of Apple: if you insist on the "very best", get an Apple and soup it up (by inserting many expensive circuit cards); the Radio Shack cannot be souped up; and although Radio Shack's programs are adequate for most purposes, you can find a greater variety of programs for the Apple than for Radio Shack.

LOW-COST COMPUTERS

If you want a low-cost computer for educational use, you have several choices. Although you can buy Radio Shack's Color Computer for $399 list ($310 at discount), an even better deal is Commodore's VIC-20, which lists for just $299.95 ($279 at discount) and is fancier than the Color Computer. The VIC-20 is a fantastic bargain, and I recommend it wholeheartedly. I'm planning to buy ten of them myself.

For an even cheaper computer, look at the Sinclair ZX-81, made in England. Today you can buy it in England; in 1982, it will be exported to the United States, where it will sell for $149.95. It isn't as nice as the VIC-20, but it's $150 cheaper. Do *not* buy its predecessor, the ZX-80, which costs more and offers less. When the ZX-81 is exported to the United States, the ZX-80 will become obsolete.

ATARI

Atari's computers are quite nice— they're much more modern in concept than the Radio Shack or Apple. Atari has the best graphics, the best music, the best editor, the best games, the best keyboard layout, and the best method of attaching extra devices easily and safely.

The Atari computer comes in two models: the 400 and the 800. The 400 is dirt cheap: Atari has recently reduced the list price to $399. The 800 is more expensive; but if you go to discount dealers, you can get the 800 more cheaply than Radio Shack's model 3, and *much* more cheaply than an Apple 2-plus.

Atari is suffering from only three problems:

The Atari 400's keyboard is a "monopanel", which means it doesn't have real keys; so you can't touch-type. (But the 800 doesn't have this problem.)

The Atari is more awkward to program than the Radio Shack and the Apple, because Atari's version of BASIC is weird. (But Atari's manufacturer plans to sell a better version of BASIC in August 1981 for $90.)

Because the Atari is relatively new, and because the manufacturer revealed only recently the full details about how the computer works, few programs are available for the Atari yet. (But Atari programmers are rapidly trying to catch up.)

I expect the Atari will ultimately be a smashing success. Though Radio Shack and Apple still have the lead in sales, Atari stands a good chance of catching up.

ZENITH AND SUPERBRAIN VERSUS RADIO SHACK'S MODEL 3

Heathkit used to make a computer, called the WH-89. Zenith bought the Heath company, so the computer is now called the "Zenith Z-89".

Zenith's Z-89, Intertec's Superbrain, and Radio Shack's model 3 all look alike. Each is a single molded unit containing a CPU, RAM, ROM, keyboard, screen, and disk drive. (The disk drive is to the right of the screen.)

The Zenith includes just *one* disk drive, and it holds very little information (single density). The Superbrain includes *two* disk drives, and each is double-density, so the Superbrain hods four times as much information as the Zenith. The Radio Shack resembles the Superbrain: two disk drives, each double-density.

The Zenith and Superbrain have fancier keyboards than Radio Shack, and show more characters on the screen simultaneously than Radio Shack.

From that description, the Superbrain seems to be the best deal. But there are three "catches":

The Superbrain costs more.

The Superbrain's price does not include BASIC, which costs extra.

The Superbrain's manufacturer, Intertec, doesn't check the computers before shipping them. There is no quality control. As a result, many of the Superbrains are "D.O.A.", Dead On Arrival: if you buy a Superbrain, there's a good chance that when you take it out of the carton it won't work. Moreover, Intertec is quite nasty about repairs: to get the computer repaired, you must pay for the shipping in both directions, even though the fault is Intertec's, not yours. (But don't give up hope. You'll probably be lucky: the Superbrain you receive will probably be one of the ones that works. And if you want to buy a Superbrain "safely", buy it from a local dealer, make the dealer check the equipment before you pay for it, and make the dealer promise to take care of all repairs.)

You'll probably prefer Radio Shack's model 3, because of its two double-density disks and low price. But if you want to buy a computer mainly for "word processing" (which means "editing manuscripts and correspondence"), you'll be a lot happer with the Zenith than with Radio Shack's model 3, because of the Zenith's nicer keyboard and screen and its ability to handle the "Magic Wand" word-processing program.

OSBORNE

The "Osborne 1" is a new computer that costs $1795 and includes a CPU, RAM, ROM, keyboard, screen, two disk drives, and lots of super-high-quality software for business applications. Because the low price inludes so much high-quality software, this computer is being highly praised throughout the computer industry. However, shipments will not begin until the fall of 1981, the disks are only single-density, and the screen will make you squint (it's tiny); so you'll want to shell out extra money, for the optional denser disks and larger screen.

LARGER COMPUTERS

Many larger computers are available also. Most of them are disgustingly boring and overpriced; but if the computers I already mentioned are too small for your needs, you're stuck and must buy a dinosaur.

Groundwork

ACQUISITION

Which computer should you buy? Before you start hunting, ask yourself this question: will a computer really help you?

At first glance, having a computer sounds nifty. But if you get a computer, you'll have lots of hassles.

REPAIRS

Since a complete computer system contains so many parts (the CPU, RAM, keyboard, disk drives, printer, etc.), *at least one* of the parts will be broken, and you'll need to get it repaired. If you get the repair done promptly (within 90 days), the manufacturer or store will pay for the repair, so you've lost nothing... except your temper.

MANUALS

Another difficulty is the manual that comes with the computer. Chances are, you won't understand it, and you'll have to ask your friends for help, or hire a consultant, or call me. (Since I'm a public-service organization, I don't charge you anything for help I give over the phone.) Of course, you can also try getting help from the company that sold you the computer; but unfortunately, most companies tend to be somewhat curt, especially if your question is long-winded.

If the salesman is candid, he'll say, "I don't know how to run the computer I sold you. If you want to learn how to run it, read the manual. No, I haven't read the manual myself; it's too long-winded and complicated and vague. If you don't like the manual, go take one of our courses. The course is expensive and won't teach you as much as you need to know, but at least will make you feel you're making *some* progress."

Most salesmen are not that candid.

PROGRAMS

If you try to write your own programs, you'll discover Murphy's law: no matter how long you think the program will take to write, it will take you longer. If you're wiser, and try to buy a pre-made program from somebody else, you'll find that the program doesn't work as well as the ad implied, the program's manual is either missing or impossible to understand, and— more important— you'll need to change the program to make it handle your own personal needs.

DATA ENTRY

Suppose you're lucky, and finally figure out how to use the program. Your next odious chore is to feed in the data that you want the program to process. Feeding data into a computer program is sheer drudgery, but you must do it (or stick it to a minimum-wage employee).

COWORKERS

Your next unhappy surprise is that your coworkers will both hate and fear the computer. You must become their teacher and psychologist.

SUMMARY

Those hassles are only the tip of the iceberg, but the message is clear: buying a computer at first seems joyfully exciting but quickly becomes nerve-racking.

Finally, however, the nerve-racking transition stage will be over, and you'll be thrilled.

Question: is the nerve-racking transition stage worth the effort? If you're going to be using the computer frequently, the answer is: yes. If you'd be using the computer only infrequently, the answer is: don't bother buying a computer; continue doing your work manually.

MINIMUM REQUIREMENTS

Let's assume you've definitely decided that a computer would be useful enough to justify the effort spent on the transition. So you've decided to buy a computer, under the conditions that (1) the computer is powerful enough to fill your needs and (2) the price is low enough to be affordable.

Which computers are powerful enough to meet your needs? The answer depends on what you want to use a computer for.

If you want a computer to help run your business, buy a computer that has at least 2 disk drives, at least an 80-column impact printer, and at least 40K of main memory. (Most microcomputers satisfy the "40K" requirement by giving you 32K of RAM plus at least 8K of ROM.) Those are the *minimum* requirements for creating a decent business system.

Here's why you need at least 2 disk drives.... To be safe, you should always make a second copy of whatever you put on a disk— in case the first copy gets accidentally destroyed. The second copy should be on a second disk. You should store the second disk far away from the first disk, to reduce the chance of both disks getting destroyed simultaneously. Your computer should include a second disk drive, to let you produce the copy easily.

You need at least 40K of main memory, because the typical business program consumes about 16K, the typical disk operating system consumes about 8K, and the typical BASIC-interpreter-and-editor-and-monitor-and-fundamental-operating-system consumes about 16K.

Again, those are the *minimum* requirements. You might need more. The first question to ask is: how many kilobytes (or megabytes) do you want to store on the disk?

For example, suppose you have 5000 customers, and for each customer you have 2 long typewritten lines of information (including name, address, phone number, amount due, etc.). Figuring that each "long typewritten line" holds 80 characters, you therefore have 160 characters per customer; multiply that by 5000 customers, and you get a grand total of 800,000 characters. Since a "character" is the same thing as a "byte", you have 800,000 bytes. Since a thousand bytes is called a "kilobyte", you have 800 kilobytes.

You can put the 800 kilobytes on one large disk. Or you can use two small disks, each containing 400 kilobytes. Or you can use ten teeny disks, each containing 80 kilobytes. The easiest choice is to use one big disk (800 kilobytes), and put it in a drive that can handle an 800-kilobyte disk; but such a drive is expensive. To save money, you can use a smaller drive instead, and pop many small disks in and out of the small drive; but popping small disks in and out requires manual labor which takes time, and also increases the probability of error: you might accidentally pop in the wrong disk at the wrong time. Also, if you use many small disks, you must make sure the program is intelligent enough to print messages on the screen that say "Thank you. Now please insert another disk."

Using many small disks is reasonably easy, if you're processing all the customers *sequentially* (starting with the first customer, and continuing to the last customer). But if you want to skip around— if you want to look at the first customer, then skip to the 3000th, then hop back to the 850th, etc.— you'll have to switch the small disks many times, which means you'll probably get confused and make an error. So for *sequential* access, small disks are okay; but for *random access* (which means you're skipping around), you'll need a single large disk drive instead.

Another question to ask is: how many of your customers are *active*? If half of your customers are dead accounts that you rarely have to worry about, put those customers on a separate disk and store it in a filing cabinet. In computer lingo, disks and tapes that are rarely used are called *archives*. If your entire data base requires 800K, but half of the customers are needed only for the archives, make your decision about "which disk drive to buy" as if you had only 400K to worry about.

I said that the printer should have at least 80 columns (i.e., it should handle paper wide enough to hold at least 80 characters across). If you want the computer to print large tables for financial analysis, 80 columns might not be enough: you might have to insist on 132 columns (i.e., 132 characters across).

A *word processor* is a computer that helps you compose and edit manuscripts and business correspondence. If you want the computer to be a word processor, you'll need a printer that has a daisy wheel or thimble or golf ball: dot-matrix is *not* acceptable for word processing, unless the dot-matrix is of unusually high quality.

Those are the issues you must face, if you want the computer to handle a small business. If you want the computer to handle a *large* business, you must face additional issues— which I'll discuss later.

If you're buying a computer just for pleasure and education, congratulations: you do *not* need disk drives, you do *not* need a printer, and you do *not* need lots of main memory. Instead, you can get just a tape recorder (which is much cheaper than a disk drive), you can rely on looking at just the video screen (instead of a printer), and you can get just 24K of main memory (16K of RAM and 8K of ROM). In fact, if you're buying a computer just to "learn about computers", you can get away with just 12K of main memory (4K of RAM and 8K of ROM); that amount of memory is too small to run any fancy programs but is nevertheless adequate for fascinating experiments; it's adequate for a beginning class in programming.

DIALECTS OF BASIC

Volumes 1 and 2 explained how to program the computer by using words such as PRINT, INPUT, and GO TO. The words explained in volumes 1 and 2 form a language called BASIC, which is part of English. If you don't want to program in BASIC, you can use different languages (such as FORTRAN, COBOL, and PASCAL), which are explained in volumes 7 and 8; but if you're buying your first computer I recommend BASIC, because it's the easiest to learn.

BASIC comes in several different dialects. The original dialect was invented at Dartmouth College in 1964. It was primitive: it didn't even include the word INPUT (you had to say READ instead), and it didn't even include string variables. Afterwards, Dartmouth rapidly improved its version of BASIC, so that today Dartmouth uses what it calls the "7th edition".

(Today, many computer manufacturers brag that their computers have "improvements over the original Dartmouth BASIC". That's not much to brag about, since even *Dartmouth* has improvements over the original Dartmouth BASIC.)

Dartmouth's original research was done on a General Electric computer. Afterwards, General Electric quit making computers and sold its computer factory to Honeywell. So today, Dartmouth's computer is called a "Honeywell".

Since much of Dartmouth's research on BASIC was funded by the National Science Foundation, the results of the research were in the public domain, and computer manufacturers could use them without having to worry about copyrights or patents. The first company to copy Dartmouth's ideas was Digital Equipment Corporation (DEC), which manufactures PDP computers. DEC put BASIC on the PDP-8, the PDP-10, the PDP-11, and the PDP-20. DEC's versions of BASIC were similar to Dartmouth's. The versions put on the PDP-8 were quite primitive (almost as bad as Dartmouth's first edition), but the versions put on the PDP-10, PDP-11, and PDP-20 were more sophisticated; and recently, DEC has put decent versions of BASIC even on the PDP-8. For the PDP-10, PDP-11, and PDP-20, the following versions of BASIC are available; I've listed them from worst to best:

WORST	BASIC	(for PDP-10)
↑	U.Penn. BASIC	(for PDP-10)
|	BASIC-11	(for PDP-11)
↓	BASIC-PLUS	(for PDP-11)
BEST	BASIC-PLUS-2	(for PDP-11 and PDP-20)

Soon after DEC started putting BASIC on its computers, Hewlett-Packard decided to do likewise: Hewlett-Packard put BASIC on the HP-2000 computer, and then put a better version of BASIC on the HP-3000 computer. Unfortunately, Hewlett-Packard's BASIC is more difficult to deal with than DEC's: on Hewlett-Packard computers, each time you use a string you must write a "DIM statement" that warns the computer how many characters will be in the string.

Later, other manufacturers decided to use BASIC also. Most manufacturers imitated Honeywell and DEC; but Data General, Wang, and IBM made the mistake of copying Hewlett-Packard instead.

That's how BASIC became popular on maxicomputers and minicomputers. The first popular *micro*computer was the Altair 8800, which used a version of BASIC invented by a 20-year-old kid named Bill Gates. His version imitated DEC's. Unfortunately, the Altair computer was manufactured by a company called "MITS", which didn't treat Bill Gates fairly, so Bill broke away from MITS and formed his own company, called Microsoft. Altogether, Microsoft has invented many versions of BASIC; the most popular have been these:

	Version of BASIC	Computers that use it
WORST	4K BASIC	obsolete computers using an 8080 CPU
↑	TI BASIC	Texas Instruments computers (TI-990 and TI-99/4)
|	8K BASIC	Sorcerer and Compucolor computers
|	6502 & 6809 BASIC	Apple, Commodore, Ohio Sci., Radio Sh. Color Computer
|	TRS-80 Level 2 BASIC	Radio Shack models 1 & 3 without disks
|	14K BASIC	obsolete computers using an 8080 CPU
↓	disk BASIC version 4	Radio Sh. models 1, 2, & 3 with disks; Zenith; old CP/M
BEST	disk BASIC version 5	new computers that use CP/M

Most of those versions require an 8080 or Z-80 CPU (exceptions: TI BASIC requires a TI-9900 CPU; 6502 BASIC requires a 6502 CPU). Most versions are intended for computers that don't have disks (exceptions: disk BASIC versions 4 & 5 require disks).

If you look at that list of who uses Microsoft BASIC, you'll see that it includes almost every microcomputer manufacturer. They've all paid Bill for permission to use his Microsoft BASIC.

Some manufacturers tried to avoid paying Bill. They were sorry!

For example, Radio Shack tried hiring somebody else to write Radio Shack's BASIC. The person who Radio Shack hired quit in the middle; so Radio Shack's original BASIC was never completed. Nicknamed "Level 1 BASIC", it's still a half-assed mess. Radio Shack, like an obedient puppy dog, then went to Bill, who finally wrote a decent version of BASIC for Radio Shack; Bill's version is called "Level 2".

The same thing happened with Apple. Apple's original attempt at BASIC was called "Apple Integer BASIC"; it couldn't handle decimal points, and made the mistake of imitating Hewlett-Packard instead of DEC. Eventually, Apple's president came to his senses and hired Bill, who wrote Apple's better BASIC, called "Applesoft" (which means "Apple BASIC by Microsoft"). That version was only for use with tape. Later, when Apple's president wanted to use disks, he made the mistake of not rehiring Bill— which is why Apple's disk system is worse than Radio Shack's.

Notice that to get Microsoft's best version (which is "disk BASIC version 5"), you must by a "new computer that uses CP/M". CP/M is a disk operating system that requires an 8080 or Z-80 CPU. The CP/M itself costs about $170; Microsoft disk BASIC version 5 costs an additional $325; so the total cost is about $495. If you say you want to buy CP/M and Microsoft BASIC, the salesman gives you two disks: one contains CP/M; the other contains Microsoft BASIC.

If you have an Apple and want to use disk BASIC version 5, send $349 to Bill's company (Microsoft), and say that you want the "Z-80 Softcard"; in return, Bill will send you a printed-circuit board ("card") that contains a Z-80 CPU, a disk that contains both CP/M and Microsoft's disk BASIC version 5, and a slightly old-fashioned disk (in case your Apple is an old model); just pop the card into the Apple, pop a disk into the drive, type the word MBASIC (which means "Microsoft BASIC), and— presto— you have disk BASIC version 5 on the Apple. Though the list price is $349, you can get a discount:

$252 from B G Sound (60 E. 9th St., New York, NY 10003, phone 212-674-1185)

$250 from Computer Exchange (Box 1380, Jacksonville, Oregon 97539, phone 503-899-7297, out-of-state 800-547-1289)

The mistakes made by Radio Shack and Apple were repeated by many other companies. For example, a computer manufacturer called "North Star" didn't want to pay Bill, and invented its own "North Star BASIC", which imitated Hewlett-Packard; needless to say, it was a disaster; so today, most people who own North Star computers are buying the CP/M operating system, to get Microsoft's disk BASIC version 5. At Atari, an executive didn't want to pay Bill, and instead came out with "Atari BASIC", which imitated Hewlett-Packard; as a result, Atari's computer never became as popular as Atari hoped, and the executive was fired, and Atari's been desperately trying ever since to get Microsoft BASIC; Atari plans to have it in August 1981. A smaller computer manufacturer, "APF", invented a computer called "The Imagination Machine", which is very inexpensive and very beautiful, except that it doesn't use Microsoft BASIC (in fact, the guy who designed the Imagination Machine's BASIC is the most incompetent designer I've ever heard of); as a result, the APF Imagination Machine is dying a slow death, in spite of APF's attempt to revive it.

Moral: if you're trying to manufacture a microcomputer, make sure you pay Bill! And if you're trying to *buy* a microcomputer, make sure it handles Microsoft BASIC.

How good is Microsoft BASIC? Is it better than DEC's BASIC? Almost. Microsoft's disk BASIC version 5 is better than DEC's BASIC-PLUS, though not as good as DEC's BASIC-PLUS-2.

Throughout that discussion, I used words such as "better" and "worse". How can one version of BASIC be "better" than another? I mean that it lets you write programs more easily. Even if you have a "worse" version of BASIC, you can still write decent programs— but writing them is more difficult and awkward.

Radio Shack

MODEL 3

Radio Shack has stopped manufacturing the model 1. Radio Shack has replaced the model 1 by the model 3, which is slightly better and slightly more expensive. For a low-cost high-quality computer, I recommend the model 3. At the moment, the model 3 suffers from only three problems, which are temporary:

Radio Shack isn't yet manufacturing enough model 3 disk drives to meet the demand.
Some of the model 1 programs haven't yet been rewritten for the model 3.
Some of the disk controllers being shipped don't work. (But the repair is fast & free.)

The model 3 comes in three versions. The cheapest version, which is junk, costs $699. It's junk because it's version of BASIC is only "Level 1", which is incomplete: it can't even alphabetize a list of three names! The next version, which is reasonable, costs $999; it includes 14K of ROM (which contains a decent version of BASIC) and 16K of user RAM. The fanciest version costs $2495; it resembles the $999 version, but adds 16K of extra RAM (making a grand total of 32K RAM), two disk drives, and an RS-232 interface card (which helps you attach the computer to a telephone).

The model 3 is packaged nicely. The entire computer system (including the Z-80 CPU, ROM, RAM, keyboard, and video screen) is a single molded unit, out of which comes a single power cord. If you buy the fancest version, the molded unit includes even the two disk drives.

Unfortunately, the price does *not* include a tape recorder (which costs $60), a printer (which costs anywhere from $219 to $1960), a printer cable (which costs $20, $29, or $39), an acoustic coupler (which attaches to a telephone and costs $199), or an RS-232 cable (which attaches the RS-232 interface card to the acoustic coupler and costs $20).

The model 3 is reliable (except for the temporary problem with initial shipments of the disk controller). Even the keyboard and tape are reliable. (The problems that occurred with the early versions of the model 1 have been eliminated.)

The model 3 improves on the model 1 in several ways. Its screen displays a greater variety of characters: capital letters, lower-case letters, the entire Greek alphabet, the entire Japanese Katakana alphabet, foreign accents (for Spanish, French, German, and Danish), all four card suits (clubs, diamonds, hearts, and spades), the male and female sex symbols, a picture of a boy, a picture of a girl, a rocket ship, a Space Invader, a huge pointing finger, a smiling face, a frowning face, plus many other symbols used by artists, scientists, business executives, and typesetters. Nifty! Each 5-inch disk holds lots of information: 256 bytes per t-sector, 18 t-sectors per track, and 40 tracks per disk, making a grand total of 184,320 bytes per disk. (The model 1 stored only 89,000 bytes per disk.) The model 3 handles tapes quickly: 1500 bits per second. (The model 1 stored only 500 bits per second.) The model 3 lets you more easily copy information from one disk to another (disk, printer, tape, screen, and main RAM). The model 3 contains many other, subtler improvements over the model 1.

Any program written for the model 1 will work on the model 3— if the program is written entirely in BASIC. But if the program uses fancier kinds of fiddling— if it uses PEEK or POKE or assembly language or machine language— it will *not* automatically work on the model 3; you must rewrite the program.

COLOR COMPUTER

If you want to buy a decent computer, but don't want to spend the $999 that Radio Shack charges for the decent model 3, get Radio Shack's new Color Computer instead. It comes in two versions. The cheaper version costs just $399, and includes a keyboard, a 6809 CPU, an 8K ROM containing BASIC, a 4K user RAM, and an RF modulator (which attaches to your home's television). The fancier version costs $599; it includes a larger ROM (16K instead of 8K) and a larger RAM (16K instead of 4K).

The computer— even the cheaper version— can draw color pictures on your television, play music, and handle practically every aspect of BASIC that I covered in volumes 1 and 2. You can attach it to a tape recorder (which costs $60), a printer ($219), joysticks ($25 per pair), and a disk drive (available after July 1981; price not yet announced). In short, it's a wonderful computer to learn on!

But since it's new, few pre-programmed tapes are available for it yet; and until the disk drive becomes available, the computer is inappropriate for business. Unfortunately, it prints only 32 characters per line (the model 3 prints 64 characters per line). And although the computer gives you a lot for your money, the new VIC-20 computer from Commodore is an even better deal.

POCKET COMPUTER

For a very low price, Radio Shack will sell you a computer in the shape of a pocket calculator. The original price was $249, but lately Radio Shack has been holding a sale: the sale price is $199. Check to see whether the sale price is still available.

Radio Shack calls it the Pocket Computer. It's manufactured for Radio Shack by Sharp (the Japanese company that also makes calculators and televisions. Sharp calls it the PC-1211. If you wish, you can buy it directly from a Sharp dealer instead of from Radio Shack. (An example of a Sharp dealer is Atlantic Northeast Marketing, Box 921, Marblehead, MA 01945, phone 617-639-0285.)

The computer is not nearly as good a deal as the Color Computer (or the VIC-20): its keyboard is too tiny for pleasant typing, its calculator-style display shows only one line of your program, and its version of BASIC lacks many important features. But you might like it anyway, if you can't afford the Color Computer (or VIC-20), or if you don't have a television, or if you need a computer that can fit in your pocket. The computer runs on tiny batteries and weighs only 6 ounces!

DISCOUNT DEALERS

In volume 3, I listed four dealers who sell Radio Shack equipment at discount: Micro Management Systems, Pan American Electronics, C & S Electronics Mart, and VR Data. Since then, the number of discount dealers has increased. Today, your "best bets" are these:

	List price	CDA	Perry	Computer Plus	Texas	Rand's
4K Color Computer	$399	$339	$339	$310	$339	$319
16K Model 3	$999	$869	$850	$859	$849	$839
2-disk Model 3	$2499	$2150	$2100	$2149	$2100	$2124
64K Model 2	$3899	$3385	$3500	$3375	$3349	$3298

Each of them— CDA, Perry, Computer Plus, Texas, and Rand's— is an authorized Radio Shack dealer. Here are their addresses, with additional comments.

Computer Discount of America, West Milford Mall, West Milford, NJ 07480.
Phone 201-728-8080, out-of-state 800-526-5313. Also sells Atari.

Perry Oil & Gas, 137 North Main St., Perry MI 48872.
Phone 517-625-4161. Free shipping for all equipment.

Computer Plus, 245A Great Rd., Littleton, MA 01460.
Phone 617-486-3193, out-of-state 800-343-8124.

Texas Computer Systems, P.O. Box 1174, Brady, TX 76825.
Phone 915-597-0673, out-of-state 800-351-1473. Free shipping for the Color Computer.

Rand's, 2185 East Fry Blvd., Sierra Vista, AZ 85635.
Phone 602-458-2477. Sells all other Radio Shack equipment at 15% off list.

NASTINESS

Radio Shack is becoming nastier. For example, if you buy a cheap Radio Shack computer and later want to improve it (by adding extra RAM or a better ROM), Radio Shack charges you for the labor of installing the new memory chips. (The labor used to be free.) If you bought one of the first model 3's (which contained an error) and want Radio Shack to correct the error, Radio Shack will charge you $20 for the improved part, plus extra for labor. (Radio Shack used to correct its own errors for free.) Radio Shack used to send a helpful newsletter to all its computer customers; but now the free subscription lasts only one year, after which you must pay $12 annually. If you buy a Radio Shack computer from an authorized mail-order discount dealer, and then take it to your *local* Radio Shack store for the free repairs that the warranty guarantees, the local store's manager will be as mean to you as possible— and Radio Shack's headquarters (in Fort Worth) approves of the meanness. If you ask Radio Shack to repair a computer, the repairman will hunt to see whether you inserted any circuitry not manufactured by Radio Shack; if he finds such circuitry, he'll remove it and charge you for the labor of removing it, even if that circuitry was not the reason why your computer needed repair, and even if that circuitry was actually *helping* the computer. In 1981, Radio Shack will refuse to sell you an expansion interface (for a model 1), unless you can prove that you bought the model 1 from a company-owned store or an *authorized* dealer.

USED COMPUTERS

If you'd like to sell or buy a used Radio Shack computer, contact TSE-Hardside (6 South St., Milford, NH 03055, phone 800-258-1790, out-of-state 603-673-5144). Hardside buys used computers for about one-third of list price, refurbishes them, and resells them for about two-thirds of list price. Hardside even has model 1 computers (which are in high demand, since they are less expensive than the model 3, and since more software has been written for the model 1 than for the model 3).

To sell or buy a used computer, try either Hardside or your local computer club or classified ads (in local and national newspapers and magazines).

IMITATIONS

A Hong Kong company (Eaca International) has been manufacturing an imitation of the model 1. The imitation is sold under various names: "TRZ-80", "PMC-80", and "Video Genie". It handles exactly the same software as the model 1. Radio Shack is suing Eaca and its distributors; but since the suit hasn't been settled yet, the imitation is still being sold.

In the United States, the distributor is Personal Micro Computers (475 Ellis St., Mountain View, CA 94043), which calls it the "PMC-80" and charges $675. But you can get it cheaper from these discount dealers (who buy from Personal Micro Computers):

$595 from Simutek Computer Products, 4877 E. Speedway Blvd., Tucson, AZ 85712 phone 602-323-9391, out-of-state 800-528-1149

$499 from The Graphic Source, Route 2, Box 44, Canton, KS 67428 phone 316-628-4935

The PMC-80's price includes the keyboard, Z-80 CPU, 12K ROM (containing Level-2 BASIC), 16K user RAM, and tape recorder. It does *not* include a screen; instead, the PMC-80 attaches to your home's television. The keyboard does *not* include a numeric keypad (but that's not important— hardly anybody uses Radio Shack's numeric keypad anyway). The keyboard has only one SHIFT key, which is on the left side; the right-hand SHIFT key is missing. The tape recorder is mounted in the keyboard, so that the entire computer is a single molded unit.

You can buy a different version— the "PMC-81"— for $740 from Personal Micro Computers. It resembles the PMC-80, but has more keys (a right-hand SHIFT key, numeric keypad, and several extra keys) instead of a tape recorder.

Personal Micro Computers sells its own expansion interface, called the EXP-100, which attaches to the PMC-80 (or PMC-81) but does *not* attach to Radio Shack's original TRS-80. The EXP-100 costs $415 (but you can get it for just $299 from The Graphic Source). Like Radio Shack's expansion interface, it includes a disk controller, Centronics parallel printer interface, and 50-wire bus; in addition, it includes an RS-232 serial interface and an S-100 bus. You can buy 32K of RAM to put into the EXP-100, for $295 ($219 from The Graphic Source).

Another imitation of the model 1 is the LNW80, manufactured by LNW Research (14661-C Myford Rd., Tustin, CA 92680, phone 714-552-8946). Unfortunately, the LNW's price rose on June 1, 1981; the new price is $1450 (plus $15 shipping). That includes a full keyboard (with numeric keypad), Z-80 CPU, 12K ROM, 48K RAM, disk controller, Centronics parallel interface, RS-232 interface, and disk operating system. For an extra $464, you also get a monitor, a disk drive, and a cable for the drive. The price does *not* include a tape recorder. Before you order, ask about the delivery time: there's a delay. The screen shows lower-case letters; it shows white-on-black and also black-on-white. LNW promises to make the screen show 24 lines of 80 characters, but that option isn't available yet (in spite of the advertisement's claim) and will cost extra.

Many companies sell disk drives that attach to Radio Shack's computers. If you want a disk drive for a model 1 computer, the cheapest sources are Access Unlimited ($295 for 40-track, $430 for 80-track, at 315 N. Shiloh, Suite D1, Garland, TX 75042, phone 214-494-0206, out-of-state 800-527-4196) and VR Data ($299 for 40-track, $419 for 80-track, phone 215-461-5300, out-of-state 800-345-8102). To use the drive, you must also buy a cable ($25 for a two-drive cable, $35 for a four-drive cable) and a disk operating system (available from Radio Shack for $15).

The disk drives in the model 3 are manufactured by Texas Peripherals, a company that's owned 50% by Radio Shack and 50% by Datapoint. To save money, buy your disk drives elsewhere. If you phone these companies, they'll tell you how to save money, by using their own disk drives instead of Texas Peripherals':

Michigan
Level IV Products, 32461 Schoolcraft Rd., Livonia, MI 48150, 313-525-6200, 800-521-3305
A.M. Electronics, 3366 Washtenaw Ave., Ann Arbor, MI 48104, 313-973-2312

Texas
Aerocomp, Redbird Airport (Building 8), P.O. Box 24829, Dallas, TX 75224, 214-337-4346
Percom Data Company, 211 North Kirby, Garland, TX 75042, 214-272-3421

Other states
VR Data, 777 Henderson Blvd., Folcroft, PA 19032, 215-461-5300, 800-345-8102
Micrcomputer Technology Inc., 3304 W. MacArthur, Santa Ana, CA 92704, 714-979-9923

Before you buy from those companies, make sure the price includes *two* disk drives and the *disk controller* and the *disk operating system (DOS)*, and ask these questions:

Is the company selling you a complete model 3 with the disk drives already installed, or is the company selling you just a disk "kit", which you must install yourself into the model 3?

If you're getting just a "kit", how hard is it to install?

If you're getting a complete model 3 with the disk drives installed already, how much RAM does the price include, and does the price include an RS-232 interface? (Most of the companies say: the price includes 48K of RAM but doesn't include an RS-232 interface. Radio Shack's $2495 list price includes only 32K of RAM but does include an RS-232 interface.)

How many tracks does the disk drive make on one side of the disk? (Radio Shack's drives make 40 tracks per side. Most of the independent companies let you choose between 40-tracks-per-side and the better-but-costlier 80-tracks-per-side.)

Do you get Radio Shack's own DOS, or do you get a slightly different DOS instead? If the DOS is different, how?

How long must you wait until you get what you ordered? (Most of the independent companies deliver quickly.)

Warning: if you insert their disk drives instead of Radio Shack's, you void Radio Shack's warranty for the rest of the computer.

Apple

PRICES

On February 1, 1981, Apple raised its prices. Here are Apple's new list prices, and the discount prices offered by mail-order dealers:

	List	Comp. Spec.	Comp. Stop	Frederick Comp. Prod.	Consumer Comp.	Computer Exchange
Apple 2-plus with 16K user RAM	$1330	$1099	$1075	$1089	$1049	$1049
Apple 2-plus with 48K user RAM	$1530	$1199	$1193	$1179	$1189	$1099
Disk drive & controller & DOS 3.3	$645	$535	$529	$535	$529	$499
Second disk drive	$525	$446	$475	$455	$445	$439

Here's how to contact those dealers:

Computer Specialties, 1251 Broadway, El Cajon, CA 92021, 714-579-0330, 800-854-2833

Computer Stop, 2545 W. 237 St., Torrance, CA 90505, 213-539-7670

Frederick Computer Products, Municipal Airport, Frederick, MD 21701, 301-694-8884

Consumer Computers Mail Order, 8314 Parkway Dr., Grossmont Shopping Center North, La Mesa, CA 92041, 714-698-8088, 800-854-6654

Computer Exchange, Box 1380, Jacksonville, OR 97530, 503-899-7297, 800-547-1289

If you buy from those discount dealers, you must add 3% for shipping. (Exception: Frederick charges only $10 to ship and Apple, and only $5 to ship a disk drive.)

IMITATIONS

Lobo sells disk drives that imitate Apple's. They work as well as Apple's, but cost less. You can get a Lobo drive for just $399 from Mark Gordon Computers (P.O. Box 77, Charlestown, MA 02129, phone 617-491-7505, out-of-state 800-343-5206).

California Computer Systems manufactures interface cards that imitate Apple's. They work even better than Apple's, yet cost less. You can buy a California card for just $115 from The CPU Shop (5 Dexter Row, Charlestown, MA 02129, phone 617-242-3361, out-of-state 800-343-6522). When ordering, say whether you want an "RS-232 serial interface" or a "Centronics parallel interface". The cable costs extra.

Epson makes a parallel interface card, for attaching Epson's printer to the Apple. The card probably handles most Centronics printers also. You can buy Epson's card for just $70 from Mark Gordon Computers (P.O. Box 77, Charlestown, MA 02129, phone 617-491-7505, out-of-state 800-343-5206). The cable costs extra.

SLOTS

If you look inside the Apple 2-plus, you'll see eight empty slots. You can buy printed-circuit boards ("cards") to put in the slots. Some of the cards are manufactured by Apple; other cards are manufactured by independents. Certain cards belong in certain slots....

In slot 0, put a card containing 16K of RAM. So if your Apple previously contained 48K of RAM, your Apple now contains a grand total of 64K of RAM.

In slot 1, put a Centronics parallel interface card. It connects your Apple to a printer.

In slot 2, put a communication card. It lets your Apple communicate with other computers.

In slot 3, put an 80-column card. It makes your Apple's screen display 80 characters per line, instead of 40. It also enables the screen to display lower-case letters, instead of just capitals. (But unfortunately, when you use an 80-column card, you can't do graphics, and you must use a black-and-white screen instead of color.)

In slot 4, put a clock. It's a card whose circuitry enables the Apple to tell you the date and time.

In slot 5, put a music synthesizer. It's a card that lets you attach the Apple to a hi-fi speaker and produce music.

In slot 6, put a disk controller. It lets you use two disk drives. (When you buy your first disk drive, make sure you get the drive itself, the controller, the cable, and the DOS.)

In slot 7, put the Z-80 Softcard. It contains a Z-80 CPU, and lets you use CP/M and Microsoft disk BASIC version 5.

Unfortunately, Apple's power supply isn't large enough to handle all eight of those cards simultaneously. If you put cards in all eight slots, the power supply will overheat. So insert just the cards that you need at the moment.

To protect the Apple from damage, turn off the power (pull the power cord out of the wall) before you insert or remove a card.

BLACK APPLES

Bell & Howell sells Apple computers to schools. But Bell & Howell's version of the Apple is black instead of yellow. Computerists have nicknamed it "The Darth Vader Apple". It's exactly the same as the Apple 2-plus, except for the color.

REPAIRS

Your Apple will probably need to be repaired frequently, for two reasons. First, you'll forget to turn the power off, before inserting or removing a card. Second, the Apple is poorly designed: it tends to overheat. The overheating occurs mainly because the Apple is inadequately ventilated: there aren't enough air vents in the case. The problem becomes particularly severe if you have at least four cards in the slots: the cards give off extra heat, and also block cooler air from circulating. The Apple gets so hot that if you touch the chips, you'll burn your finger.

To avoid overheating, buy a tiny fan, and put it inside the Apple. Put the fan at the right side, so that it blows air across the chips, from right to left.

To get your Apple repaired during the 90-day warranty period, take it to *any* authorized Apple dealer. Apple Computer Company will pay that dealer for the parts and labor.

APPLE 3

When the rumor mill began saying that Apple Computer Company would invent an "Apple 3", everybody got excited! But when the Apple 3 finally arrived, it was disappointing.

It was overpriced. It cost about $4,000, and didn't even include two disk drives: it included only one 5" floppy drive. Though its color graphics, keyboard, operating system, and version of BASIC are slightly better than the Apple 2's, those nice touches aren't enough to justify a price as high as $4,000.

Don't buy it, unless you need fancy graphics or its ability to hold 128K of RAM.

MISTAKES

Apple's biggest mistake was to sink so much effort into the overpriced Apple 3, which doesn't work. The company should have sunk more effort into low-cost computers, priced near the Apple 2-plus. The Apple 2-plus is obsolescent: although its sales are still strong, it'll become archaic in a few years, unless Apple starts putting more effort into research and development. At the moment, Apple is succeeding because of Radio Shack's incompetence; but once Radio Shack gets its own act together (which will happen in April 1981), Apple will feel the pinch, since the Apple 2-plus is a bit overpriced relative to Radio Shack. Japanese manufacturers are planning to invade the U.S. with computers similar to the Apple and the TRS-80: that invasion will hurt Apple further.

Many dealers are starting to dislike Apple. Apple's discounts to dealers have become slimmer. Apple is forcing dealers to sign contracts in which the dealer agrees not to display games (the Apple is supposed to be only for business) and in which the dealer agrees not to give customers substantial discounts. Apple is trying to put mail-order dealers out of business. Apple gave all its own employees a two-week vacation during Christmas: if a dealer tried to call Apple during those two weeks (to place an order, or to ask for help), the phone would ring repeatedly, and nobody would ever answer (not even an answering machine). When Apple received a shipment of substandard disk drives from the drive manufacturer (Shugart), Apple passed them on to dealers anyway; as a result, many customers had trouble with disk drives.

Apple has been running advertisements in which it falsely claims that the Apple was the first pesonal computer ever invented. Apple has been encouraging customers to buy the "Language Card and PASCAL", even though PASCAL offers few real advantages to most of Apple's customers. When Apple decided to "go public" and sell shares of Apple stock, the "stock offering" broke so many rules of good business practice that Massachusetts declared the sale of Apple stock to be *illegal*, on *three* grounds, and told naive investors to stay away from Apple's inflated prices.

Unfortunately, Apple is becoming a stodgy, nasty, less-than-honest bureaucracy, a far cry from the exciting days when it operated out of its inventors' garage.

Commodore

PRICES

Commodore is lowering its prices. As of July 1, 1981, the prices are:

	Model #	List	MBC	CMO
computer with 5K user RAM & 22-column display	VIC-20	$300	$290	$279
computer with 16K user RAM & 40-column display	4016	$995	$859	$850
computer with 32K user RAM & 40-column display	4032	$1295	$1090	$1080
computer with 32K user RAM & 80-column display	8032	$1495	$1335	$1279
cassette recorder & its cable	C2N	$95	$89	$87
drive holding two 176K disks & controller	4040	$1295	$1090	$999
drive holding two 533K disks & controller	8050	$1795	$1599	$1449
tractor-feed printer	4022	$795	$749	$669
cable for first peripheral (disk drive or printer)		$39	$29	$37
cable for additional peripheral (disk drive or printer)		$49	$34	$46

Warning: those peripherals (cassette recorder, drive, printer, and cable) do *not* attach easily to the smallest computer (the one that has 5K user RAM & 22-column display).

You can get the MBC price from Multi-Business Computer Systems (28 Marlborough St., Portland, CT 06480, phone 203-342-2747). You can get the CMO price from Computer Mail Order (501 E. Third St., Williamsport, PA 17701, phone 717-323-7921, out-of-state 800-233-8950).

DISK DRIVES

The "drive holding two 176K disks" is manufactured by Shugart. The "drive holding two 533K disks" is manufactured by Micropolis. So Commodore buys from Shugart and Micropolis.

In each drive, each t-sector contains 256 bytes.

In the Shugart drive, each 5" disk contains 35 tracks. The innermost track contains 17 t-sectors; the outermost track contains 21 t-sectors; the average track contains about 20 t-sectors. Altogether, the disk contains 690 t-sectors.

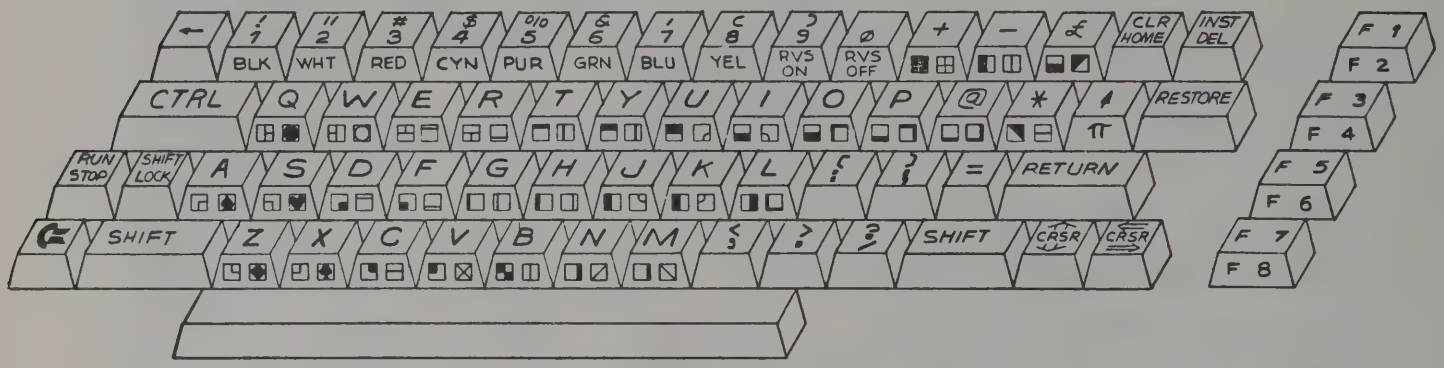
In the Micropolis drive, each 5" disk contains 77 tracks. The innermost track contains 23 t-sectors; the outermost track contains 29 t-sectors; the average track contains about 27 t-sectors. Altogether, the disk contains 2083 t-sectors.

VIC

Commodore has invented a new low-cost computer, called the VIC. It's excellent, and it costs "less than $300". (It lists for $299.95; the CMO price is $279.)

For that price, you get a keyboard (containing a 6502 CPU, some ROM, and 5K of RAM) and an RF modulator. You attach it to your home's television set. It can draw pictures in 8 colors, and play music by using 255 pitches and 4 voices.

The keyboard is full-sized, excellent, and fancy:



A typical key has a capital letter and two graphics symbols. If you press the key, you normally get the capital letter; if you hold down the SHIFT key, you get the rightmost graphics symbol instead; if you hold down the "Commodore" key (which is left of the left SHIFT key), you get the leftmost graphics symbol instead. If you hold down both the SHIFT key and the Commodore key simultaneously, you switch the computer into a totally different mode of operation— in which hitting a key normally gives you a *lower*-case letter, holding down the SHIFT key gives you a capital letter, and holding down the Commodore key gives you the leftmost graphics character.

If you wish to attach a tape recorder, you must buy Commodore's, which costs $75. The VIC uses the same version of BASIC as Commodore's slightly more expensive computer (the PET), and the VIC's tape recorder uses the same coding technique as the PET's. So if you've written a program for the PET, and you want to run the program on the VIC-20, just feed the program from the PET into the PET's tape recorder, then put that tape into the VIC's tape recorder, and— presto!— the PET program will work on the VIC, if the program is written entirely in BASIC (instead of in machine language) and provided that you realize the VIC displays only 22 characters across each line (the PET displays 40).

If you want the VIC to do fancy high-resolution plotting, attach the "SuperExpander Cartridge"; it contains extra ROM and 3K of extra RAM. It will cost $100 or $125; Commodore hasn't decided on the exact price yet.

Like the PET, the VIC contains a decent version of BASIC and a top-notch editor. Because of its low price, I recommend it highly. In fact, it's the most cost-effective computer ever invented.

I've been describing the VIC-20, which is available now. (I own one myself.) A fancier version, called the VIC-40, will become available later.

NEW ADDRESS

Commodore has moved its U.S. headquarters from California to Pennsylvania. The new address is 950 Rittenhouse Rd., Norristown, PA 19403, phone 215-666-7950.

Which system to buy

RADIO SHACK'S MODEL 3

To begin your search for a low-cost high-quality computer, look at Radio Shack's model 3. It uses a good version of BASIC, can handle disks, and costs little.

Its version of BASIC is better than Apple's, its disks hold more than Apple's, and it costs less than Apple. Radio Shack's fanciest version— which lists for $2495 and is available at discount for $2100— even includes a disk controller, two double-density disk drives, a 12" monitor, a full keyboard (upper-case, lower-case, Japanese, Greek, graphics characters, and numeric keypad), a Centronics parallel printer interface card, an RS-232 serial communications interface card, and a clock. To get an Apple with all those features would cost far more!

If possible, buy that fancy version. If you can't afford it, get the next smaller version, which lists for $999 and is available at discount for $839.

VIC

If you can't afford even $839, get the Commodore VIC-20, available at discount for $279. It gives you more bang-per-buck than any other computer being sold today. Terrific!

SINCLAIR & THE POCKET COMPUTER

Suppose you can't afford even the VIC, and you're unwilling to wait (until you've saved enough money for the VIC, or until Sinclair's ZX-81 comes out). In that case, you're stuck: your only choices are Sinclair's ZX-80 and Radio Shack's Pocket Computer.

Neither is American. Radio Shack's Pocket Computer is manufactured by Sharp in Japan, where it sells for $125; in the United States, Radio Shack and Sharp both sell it for $249. Sinclair's ZX-80 computer is manufactured in England; you can get it in the United States for $199.95 (plus $5 shipping) from Sinclair Research's warehouse (1 Sinclair Plaza, Nashua, NH 03061, phone 800-543-3000 and ask for operator 508, unless you live in Ohio, where you must phone 582-1364).

Can't afford even $199.95? Then get the ZX-80 as a kit, and assemble it yourself. The kit costs just $149.95, from Micro-Ace (1348 E. Edinger, Santa Ana, CA 92705, phone 714-547-2526, free shipping). The kit requires soldering and 3 hours of labor.

The Sinclair and Radio Shack's Pocket Computer are both cleverly designed; but to make the price so low, the designers took many short-cuts. The hardware and software are both substandard. So I recommend that you continue saving your pennies, and get the VIC (or the Radio Shack model 3). But if you can't afford even the VIC, the Sinclair and the Radio Shack Pocket Computer are your only choices.

The Sinclair's major defects are the small RAM (only 1K), the inadequate version of BASIC (no decimals, no arrays of strings, no DATA statement, and a too-limited version of the IF...THEN statement), and the awkward keyboard (it's a monopanel). The Radio Shack's major defects are the small screen (only one line, which is only 24 characters wide), the inadequate version of BASIC (difficulties in handling strings, and no DATA statement), and the awkward keyboard (the keys are too small).

You can soup up the Sinclair. For an extra $99.95 (plus $4 shipping), you can buy an extra 16K RAM. For an extra $39.95 (plus $2.50 shipping), you can buy an 8K ROM that contains a better version of BASIC (it has decimals and has arrays of strings, though it still lacks a DATA statement and still has a poor version of the IF...THEN statement). But instead of sinking all that money into the ZX-80, get the VIC— or wait for the ZX-81, which is better than the ZX-81 and less expensive, but so far is available only in England. It will cost $149.95, and be slightly better than the ZX-81 with 8K ROM (which today costs a total of $239.90).

APPLE

The Apple 2-plus costs more than the Radio Shack model 3. And the Apple 2-plus is *worse* than the Radio Shack model 3, in general. But in a few ways, the Apple 2-plus is better than the Radio Shack model 3: the Apple 2-plus is prettier (beige looks better than silver); the Apple 2-plus can draw high-resolution color graphics (the Radio Shack draws only black-and-white and only with low resolution); the Apple 2-plus lets you insert cards in its slots (although the cards are expensive); more software is available for the Apple 2-plus than for the Radio Shack (until programmers finish converting programs from Radio Shack's model 1 to Radio Shack's model 3).

ATARI

Atari has fancier graphics than any of the other popular computers. It's *much* fancier than the Apple!

Atari also has the fanciest music (unless you insert a high-priced card into the Apple).

Atari also has the best program editor: if you write a program for the Atari, you can edit the program more easily than on any other computer. (Commodore is in second place, Radio Shack and S-100 bus computers are tied for third place, and the Apple is a distant fourth.)

Atari's disk operating system is easier to learn how to use than any other computer's.

Atari has the best computer game ever invented: it's called "Star Raiders", and it's available *only* on the Atari.

Atari's computer comes in two versions: the model 400 and the model 800. The model 400 is much cheaper than the 800. The only important disadvantages of the model 400 are its monopanel keyboard (instead of real keys) and its inability to expand (it contains few slots for add-ons). Because the model 400 costs so much less than the 800, I recommend that you get the 400— unless you can't stand the 400's monopanel keyboard. (Personally, I can't stand the monopanel, which is the main reason why I bought the 800 instead.)

In May, 1981, Atari dramatically lowered its prices. Now the Atari 400 lists for $399; that price includes 16K of RAM, but does *not* include the BASIC ROM ($60) and does *not* include the tape recorder. The Atari 800 lists for $1080; that price includes 32K of RAM and includes the BASIC ROM, but does not include the tape recorder.

For discounts, contact Computer Mail Order, which sells the $399 item for $349, the $60 item for $46, and the $1080 item for $759. (Computer Mail Order is at 501 E. Third St., Williamsport, PA 17701, phone 717-323-7921, out-of-state 800-233-8950.)

Many times, Atari advertised that new software would be available by a certain date. Each time, the software was *not* available by the advertised date. So if Atari says something will be available "soon", don't believe it. I've been waiting many months— and in some cases, several *years*— for the following Atari software, which still doesn't exist, in spite of promises: Microsoft BASIC, PASCAL, word-processing, and an accounting package. But if Microsoft BASIC is ever completed (Atari claims August 1981), it will be really nice, because Atari says Microsoft BASIC will include commands that handle super-advanced graphics concepts, including even "players and missiles". (The current version of BASIC, which is not by Microsoft, cannot do super-advanced graphics, and cannot handle strings easily (it requires you to DIM each string), and cannot handle string arrays.

INTECOLOR

The Compucolor is no longer being manufactured. But you can buy a similar computer, called the Intecolor, which is manufactured by the same company. The Intecolor is better than the Compucolor, but is also more expensive.

If you live in New England, the best way to buy an Intecolor is from Rick Kane, a dealer who happens to also be a physics professor at the University of Massachusetts (Harbor Campus). For best results, contact him at Mede's (81 Bromfield St., Watertown, MA 02172, phone 617-923-0728 weekdays 8AM-9:30AM).

DEAD COMPUTERS

Avoid APF's Imagination Machine, Mattel's Intellivision, Exidy's Sorcerer, Texas Instruments, and Ohio Scientific.

APF's Imagination Machine has the worst version of BASIC ever invented: it makes many errors, and even prints "answers" that are numerically incorrect. The list price used to be $599; recently, it dropped to $399 (because of competition from the VIC-20 and the Radio Shack Color Computer); but even at the low price of $399, I don't recommend it. APF promises to have a better version of BASIC, but when? APF says, "in a month". That answer sounds reasonable— except when you realize that APF has been saying "in a month" for the last two years. (I phone APF often, just to see whether APF's answer ever changes. It doesn't.) APF is a classic example of a computer liar.

Mattel's game machine is not programmable, unless you buy the BASIC expansion, which is ridiculously expensive and not available yet, in spite of Mattel's promises during the past several years. Like APF, Mattel is an example of a computer liar.

Most dealers have stopped selling Exidy's Sorcerer. Although it had high-resolution graphics, it lacked color, a coordinate system, disk-based Microsoft BASIC, and— most important— quality control. Also, it was overpriced.

When the Texas Instruments 99/4 came out, everybody in the computer industry scorned it, for three reasons: it was overpriced; its version of BASIC was terrible (Microsoft's worst); and its keyboard was ridiculous (the keys were too small, the keys were in the wrong places on the keyboard, and you couldn't press a key unless you released the previous key *completely*). To make matters even more ridiculous, Texas Instruments *raised* its price further. The microcomputer industry's newspaper, *Infoworld*, ran a front-page story entitled "The TI-99/4: Mistakes that only TI could afford". Recently, TI has relented: it has lowered the price, offered a better version of BASIC (as an extra-cost option), and offered a better keyboard (called the 99/4A). But those improvements are too little, too late. Programmers have decided to ignore Texas Instruments, and are writing programs for other computers instead, such as the Apple.

Ohio Scientific is no longer an independent company: it's now part of M/A-COM. Although Ohio Scientific used to have low prices, its prices have recently been less competitive, and the quality of the software and manuals is still abysmal.

HP-85

The HP-85 is Hewlett-Packard's most popular "personal" computer. Engineers who travel like it, because it's portable. It's very small and very light and nicely packaged: in one molded unit, you get the CPU, RAM, ROM, keyboard, tiny CRT, small thermal printer, and digital cartridge tape drive (which is faster and more accurate than an audio cassette drive). It attaches easily to scientific lab equipment, since most scientific lab equipment is made by Hewlett-Packard. It's the only microcomputer whose version of BASIC includes commands for graphing curves, drawing axes, labeling the axes, and scaling the graph to fit on the screen. If you press the COPY button, the computer will copy the graph from the screen to the paper.

But the price is high: it lists for $3250. If you gave me $3250, I would *not* buy an HP-85. For $3250, I could buy a much fancier computer, that has a larger screen, that can handle regular paper, that includes several disk drives (instead of a cassette), and that has a version of BASIC more suitable for handling strings. (Hewlett-Packard's BASIC forces you to say DIM for every string.) But if you don't care about price (because you got a grant from the U.S. government), go ahead and buy the HP-85. It's cute.

Instead of paying $3250, you can get it for just $2795 from Personal Computer Systems (609 Butternut St., Syracuse, NY 13208, phone 315-478-6800).

Recently, Hewlett-Packard has introduced the HP-83. It resembles the HP-85, but lacks the printer and tape drive, and lists for just $2250 ($1895 from Personal Computer Systems).

80-COLUMN COMPUTERS

Most low-cost computers display only 40 or 64 characters per line. If you insist on getting a computer that displays a full 80 characters per line, here are your least expensive options:

Radio Shack's TRS-80 model 2
Intertec's Superbrain
Vector's VIP
Zenith's Z-89 (which is the same as Heathkit's WH-89)
North Star's Horizon, plus a terminal
Commodore's 8032, plus a disk drive
an Osborne-1, plus an 80-column monitor
NEC's PC-8001, plus a monitor and a disk drive
an Apple 2-plus, with an 80-column card inserted, plus a monitor and a disk drive

Each of those costs between $2000 and $4000. Have fun shopping!

For a 25% discount on Intertec's Superbrain and North Star's Horizon, contact American Square Computers (Kivett Dr., Jamestown, NC 27282, phone 919-889-4577 or 919-883-1105). For a 25% discount on Zenith's Z-89, contact AB Computers (115 E. Stump Rd., Montgomeryville, PA 18936, phone 215-699-5826).

When a dealer gives you a price, find out what the price includes. Ask these questions:

How many K of RAM?
What size disk? How many K per disk? How many disk drives?
What about the keyboard, video, and connecting cables?
Does the price include the CP/M operating system?
Is BASIC included? Is it Microsoft BASIC? Is it version 5?
Is shipping free?
Did the dealer test the computer, to make sure it's working?
If the computer breaks, can the dealer repair it?
Where is the nearest repair center?
How long does a repair typically take?

MULTI-USER SYSTEMS

How can you let several people share the same data simultaneously?

If the group of people is small, the cheapest solution is to buy several small computers, each having its own disk drive. Into each disk drive, put a copy of the data disk. That's simple. But if you want to update the data, you must make sure to update *all* copies of the disk.

Another approach is to create a network, in which several computers all get information from one disk. If you want to do that by hooking together Apple or Radio Shack computers, contact Corvus (900 S. Winchester Blvd., San Jose, CA 95128, phone 408-246-0461); its Constellation system can attach 64 personal computers simultaneously to a hard disk.

The last approach is to buy a computer that has a hard disk and attach many terminals to the computer. To do that inexpensively, buy a microcomputer with a hard disk and the MP/M operating system. (MP/M is a souped-up version of CP/M. It can handle many terminals simultaneously.) If you're willing to spend more money, get a minicomputer or maxicomputer (instead of a microcomputer having MP/M).

The pleasantest maxicomputer is manufactured by Digital Equipment Corporation (DEC), and is called the PDP-20 (or DEC-20 or Decsystem-20). If you can't afford it, get one of DEC's minicomputers instead. DEC's newest (and most cost-effective) minicomputers are the VAX-11/580 (which is slightly smaller and a lot cheaper than the old VAX-11/780) and the PDP-11/44 (which is slightly smaller and a lot cheaper than the old PDP-11/70). If you're planning to buy a DEC minicomputer and want to use BASIC, buy DEC's best version of BASIC (which is called BASIC-PLUS-2) or at least DEC's second-best version (which is called BASIC-PLUS); don't settle for anything less. (DEC's smallest PDP-11 computers, which use the RT-11 operating system, can't handle even BASIC-PLUS.)

DEC's computers are "pleasant", but aren't necessarily "the best buy". Other manufacturers are less expensive, and might be adequate for your needs. IBM is *more* expensive and has an ugly operating system; but if you need high-speed hardware and elaborate software, you have practically no choice: you must buy from either IBM or an off-beat manufacturer or a *plug-compatible manufacturer (PCM; it's a manufacturer that makes computers that exactly imitate IBM computers but cost less)*.

When buying a big multi-user system, ask these questions:

Hardware

How much RAM and disk does the price include?
Does the price include the terminals, and the cables and controllers they require?
Does the price include the printer (and its controller)? How fast is it?
Does the price include a tape drive? How fast is it?
Does the computer have enough speed (and RAM) to handle simultaneous users quickly? How long will the average user have to wait for a respose?

Software

Does the price include the operating system?
Does the price include languages (BASIC, FORTRAN, COBOL, PASCAL, and others)?
Which *dialect* of BASIC (or other language)?
Does the price include applications packages (such as payroll and inventory)?
What's the monthly license fee, for permission to use all that software?

Environment

Does the computer require air conditioning?
How many watts of electricity does the computer require?

Repairs

How necessary is a service contract? How often will the computer need repairs?
What's the monthly fee for a service contract?
If the computer breaks, how long will the repairman take to arrive?

Satisfaction

If the computer proves unsatisfactory during the first week, can it be returned?
What are the phone numbers of people who are already happily using such a computer?

Printers

DEVILISH DOZEN

During the past year, many new printers have been invented. A dozen printers are particularly interesting, because of their high quality and low cost. Here they are, beginning with the cheapest....

SINCLAIR
about $100

Sinclair will sell a printer for $99.95, as soon as it gets FCC approval. That's the cheapest printer I've ever heard of. It has several limitations: it requires thermal paper, 5½" wide; it prints only 32 characters per line; it can't print lower-case letters (it prints only capitals); it prints very slowly (only 22 cps); and it attaches only to Sinclair's computers.

SHARP
about $150

For just $150, you can buy Sharp's model CE-122, which is a machine that performs *two* functions: it's a printer, but it's also a cassette interface. It prints on a tiny roll of adding-machine tape: the paper's width is slightly less than two inches; it prints only 16 characters per line. It can't print lower-case letters.

To use it, you must buy Sharp's PC-1211 Pocket Computer (or Radio Shack's version of it). You probably won't like the Pocket Computer, and therefore probably won't buy this printer. But the printer is historically important, for several reasons:

Its price is low.
You can buy it today.
It overcomes the major objection to the Pocket Computer (which is the inability to see the entire program at once).

If you buy the printer, you won't have to buy Radio Shack's "cassette interface", since the printer *includes* a cassette interface.

Instead of paying the $150 list price, you can get it for just $135 from Atlantic Northeast Marketing (Box 921, Marblehead, MA 01945, phone 617-639-0285); the shipping is free.

QUICK PRINTER 2
about $200

Radio Shack's "Quick Printer 2" has several advantages over the Sinclair and Sharp....

It connects to practically any computer. It can handle both popular methods of connection— "Centronics parallel" and "RS-232 serial"— and it can also handle a third method: the Radio Shack model 1 bus.

It can print lower-case letters.

It can print 32 characters per line. It prints quickly: 2 lines per second (120 lines per minute).

Unfortunately, it has one *dis*advantage: it requires aluminized paper (which looks silvery). The aluminized paper is easy to read, and the printing is neat and crisp; but if you insist on paper that's white, the Quick Printer 2 is not for you— unless you make a photocopy. (When you make a photocopy of "silver", you get white.)

The Quick Printer 2 lists for $219, but you can get it for $187 from Computer Discount of America (W. Milford Mall, W. Milford, NJ 07480, phone 201-728-8080, out-of-state 800-526-5313).

BYTEWRITER
about $300

The previous printers— the Sinclair, the Sharp, and the Quick Printer 2— are just "toys": to be considered "serious", a printer must be able to print 80 characters per line. The cheapest printer that prints 80 characters per line is the Bytewriter-1, made by Microtek. It costs just $299.

It has everything you expect of a standard printer, though it has few "extras". Like a typewriter, it uses plain paper without holes ("friction feed"), accepts paper up to 9½ inches wide, and prints 10 characters per inch, up to 80 character per line. It prints the 80 characters in 1 second. To maintain that speed continuously (80 characters per second, 1 line per second, 60 lines per minute), it prints bi-directionally. It uses a Centronics parallel interface (which means it attaches easily to most microcomputers, but does *not* attach easily to the Zenith or Superbrain, which lack the Centronics interface).

To get it for $299, contact the manufacturer (Microtek, 9514 Chesapeake Dr., San Diego, CA 92123, phone 714-278-0633, out-of-state 800-854-1081).

SEIKOSHA
about $350

The Seikosha printer is made in Japan by the same company that makes Seiko watches. The Seikosha printer resembles the Bytewriter but has an advantage: it can print high-resolution graphics. That means you can print a dot anywhere on the page. By printing dots everywhere on the page, you can make the entire page black. (But if you make the entire page black, the printer will overheat— unless you give the printer a "rest" in the middle of the job.)

So for graphics, the Seikosha is better than the Bytewriter. But unfortunately, the Seikosha is slower than the Bytewriter: the Seikosha prints only 30 characters per second, and only unidirectionally (which means that at the end of each line, you must wait for the print head to return to the left margin).

The Seikosha uses "tractor" feed. That means it requires paper that has holes down the sides, and you can adjust the tractors horizontally to accomodate paper of different widths.

The Seikosha comes in two models. The "wide" model accepts paper up to 9½ inches wide; the "narrow" model insists that the paper be 8 inches or narrower. The wide model makes each character quite wide (10 characters per inch); the narrow model makes each character narrow (12 characters per inch). No matter which model you buy, you get 80 characters per line. You'll probably prefer the wide model, since narrow paper is hard to get, and since the wide model comes with a better instruction manual.

Each model also permits "expanded" print, in which each character is twice as wide as normal. That expanded print is good for printing headings.

For the wide model, the U.S. distributor is Radio Shack, which calls it the "Line Printer 7". For the narrow model, the U.S. distributor is Axiom, which calls it the "GP-80M".

The list price (for each model) is $399. Discounts are available: you can get the wide model for $339 from Computer Discount of America (W. Milford Mall, W. Milford, NJ 07480, 201-728-8080, 800-526-5313); you can get the narrow model for $319 from The CPU Shop (5 Dexter Row, Charlestown, MA 02129, 617-242-3361, 800-343-6522).

EPSON
about $500

The same Japanese company that makes Seiko watches and the Seikosha printer also makes the "Epson" printer, which is better than the Seikosha printer.

The Epson printer comes in six models. No particular model is the "best", and no particular model is the "worst": each has its own peculiar advantages and disadvantages. Here they are:

	List price	Dot matrix, for text	Maximum paper width	Can do "friction feed"?	Speed, char. per sec.	Bidirectional?	Maximum char. per line	Typestyles	Pixels per inch, horiz.	Pixels per inch, vert.
Seikosha GP-80	$399	5×7	8"	no	30cps	no	80	2	72	63
Seikosha Line Printer 7	$399	5×7	9½"	no	30cps	no	80	2	60	63
Epson MX-70	$449	5×7	9½"	no	80cps	no	80	2	60	72
Epson MX-80	$645	9×9	9½"	no	80cps	yes	132	12	33	18
Epson MX-80 with graphics	$740	9×9	9½"	no	80cps	yes	132	24	120	216
Epson MX-80 F/T	$745	9×9	9½"	yes	80cps	yes	132	12	33	18
Epson MX-80 F/T w. graphics	$840	9×9	9½"	yes	80cps	yes	132	24	120	216
Epson MX-100	$995	9×9	15"	yes	80cps	yes	233	12	120	72

The "list price" is misleading, since most dealers sell below list. The Seikosha printers, which list for $399, you can get for $339 (or $337). The Epson MX-70, which lists for $449, usually sells for $425, and some dealers go even lower:

$390 from A B Computers (252 Bethlehem Pike, Colmar, PA 18915, 215-822-7727)
$378 from Microhouse (Box 498F, Bethlehem, PA 18016, 215-868-8219)
$360 from Geo-Computers (P.O. Box 3584, Diamond Farm Br., Gaithersburg, MD 20760, 301-840-1506)

The Epson MX-80, which lists for $645, usually sells for $499, and some dealers go even lower:

$479 from Morningside Computing (625 W. 130th St., New York, NY 10027, 212-864-0151)
$479 from Microhouse (Box 498F, Bethlehem, PA 18016, 215-868-8219)
$469 from The Graphic Source (Route 2, Box 44, Canton, KS 67428, 316-628-4935)
$462 from B G Sound (60 E. 9th St., New York, NY 10003, 212-674-1185)

The "dot matrix, for text" indicates how accurately the characters are formed. The expensive models use a high-quality matrix (9×9); the cheaper models use 5×7. Although 5×7 is legible, 9×9 looks much better, especially for the five letters of the alphabet that hang below the line ("g", "j", "p", "q", and "y").

The "maximum paper width" includes the space taken by the margins and the holes in the side of the paper. Each printer has movable tractors, so you can use narrower paper also.

Each printer can do "tractor feed", but only the most expensive can do "friction feed" also.

"Speed" gives the number of characters per second, in theory. (In practice, the printer sometimes goes slower. And the printer is very slow at "jerking the paper up", to print a new line.)

The cheapest printers are unidirectional, so at the end of each line you have to wait for the print head to return to the left margin. The more expensive printers are "bidirectional".

The most expensive printers can fit many "characters per line", because they accept wide paper and make the characters thinner.

Some of the printers have many "typestyles", including boldface, italic, "condensed" (which means narrow), and "expanded" (which means wide).

Each printer lets you draw pictures. Each picture is composed of tiny "pixels"; each pixel is a blob that's either black or white. The most expensive printers draw the pictures super-accurately, by using pixels that are very tiny— many "pixels per inch", horizontally and vertically.

Epson printers are reliable— they hardly ever need repair. And their printing is neat, crisp, sharp. (Other printers, such as the Seikosha, tend to smudge, and their printing looks "muddy".)

If you're puzzled about which printer to buy, I recommend that you begin by looking at Epson's, because of their high reliability, beautiful printing, and low price. The MX-80 is the most popular, because it's the lowest-cost printer you can buy that decently prints the letters "g", "j", "p", "q", and "y". At the discount price of $499 (or even less), it's the best bargain in the printer industry today.

CENTRONICS

about $700

Centronics is a New Hampshire company that rapidly became rich, by being the first company to make high-quality low-cost computers. Centronics manufactured every printer that Radio Shack sold (except for Radio Shack's "Screen Printer", which was manufactured elsewhere, unpopular, and quickly discontinued).

But during the past year, Centronics has been having a hard time, because of competition from the Japanese (especially Epson). Today, Centronics' printers seem over-priced. Radio Shack is starting to buy from other manufacturers instead.

But one of Centronics' printers still appeals: it's the model 737. It's the lowest-priced printer you can buy that has proportional spacing. Its dot matrix has extremely high resolution, which allows the shapes of the letters to be fancier than other printers (such as Epson's).

Centronics' quality control is inadequate: too often, when a computer store unpacks a 737 from its carton, the store finds that the 737 doesn't work. And even if you're lucky enough to get a 737 that *does* work, you must face the problem that the "friction feed" is poor: since Centronics uses a low-quality platen (plastic instead of rubber), the paper tends to slip, especially near the bottom of the page. If you decide to use the "pin feed" instead, you must buy paper that has holes in it and that's exactly $9\frac{1}{2}$ inches wide; the pins *cannot* be adjusted to fit narrower paper.

The 737 can print using either proportional spacing or monospacing. Though proportional spacing looks pretty, most "word-processing programs" don't contain commands that tell the 737 to use proportional spacing; so you wind up with monospacing instead. The Centronics' monospacing looks junky, because its dot matrix is only 8×7 (instead of the Epson's 9×9).

In short, the 737 does a nice job of printing, *if* you get a model that works, and *if* your paper doesn't slip, and *if* your word-processing program is smart enough to know how to request proportional spacing.

Radio Shack calls it the "Line Printer 4" and charges $999, but you can get it at discount:

just $737 from Micro-Management Systems (Downtown Plaza Shopping Center, 115 C Second Ave. S.W., Cairo, GA 31728, 912-377-7120, 800-841-0860)

just $725 from A B Computers (252 Bethlehem Pike, Colmar, PA 18915, 215-822-7727)

just $700 from B G Sound (60 E. 9th St., New York, NY 10003, 212-674-1185)

Centronics is coming out with a souped-up version, called the 739, which includes graphics.

DYNATYPER
about $800

The Dynatyper is a hood that fits over a typewriter's keyboard. The hood contains many plastic fingers, which press the keys. The plastic fingers are soft, so they don't hurt the keys. Installation and removal are easy: you can put the hood on or off of your typewriter within a few seconds.

The hood comes in two models. One model is for Selectric typewriters (which have a curved keyboard). The other model is for other typewriters (which have a keyboard that slants), and you must tell the Dynatyper's manufacturer exactly which typewriter you own, so the manufacturer will send you a Dynatyper having the "backspace-key finger" in the correct position.

The hood costs just $499, so the total cost of the printing system is "$499 plus the cost of your typewriter". If you own a typewriter already, your cost to computerize it is just $499.

To use the Dynatyper, you must either run a special program or buy a special box.

The "special program" (which is called a *printer driver*) is free, but is available only for the Apple, Commodore, and Radio Shack models 1 and 3. It works fine if you're using BASIC; but if you're *not* using BASIC (for example, if you're using a word processor), you must modify your computer's software. The Dynatyper's manufacturer will sell you modifications for all the popular Apple and Radio Shack software.

The "special box" is more flexible than the "special program". The "special box" works with any computer that has an RS-232 interface. It does *not* require you to modify software. The box (called the Dynastar) costs just $200 if you buy it at the same time as the Dynatyper. The box contains a 4K ROM and 2K RAM.

When using the Dynatyper, you can select any speed from 1 cps to 50 cps. To prevent your typewriter from overheating and its keys from jamming, choose just 12 or 13 cps for an IBM Selectric, and choose just 8 or 9 cps for a Smith-Corona portable.

Get the Dynatyper from the manufacturer, Rochester Data (3000 Winston Rd. South, Rochester, NY 14623, 716-244-7804).

STARWRITER
about $1500

The cheapest daisy-wheel printer you can buy is made in Japan, by Tokyo Electric. For the U.S. market, there are two distributors: C. Itoh (an international export-import company), which calls the printer the "Starwriter"; and Vista, which calls the printer the "V300".

You can buy two models. The cheaper model goes 25 cps; the faster model goes 45 cps. You can get them at discount:

$1575 for 25 cps, $1849 for 45 cps, from The CPU Shop
(5 Dexter Row, Charlestown, MA 02129, 617-242-3361, 800-343-6522)

$1495 for 25 cps, $1799 for 45 cps, from Computer Mail Order
(501 E. Third St., Williamsport, PA 17701, 717-323-7921)

$1445 for 25 cps, from A B Computers
(252 Bethlehem Pike, Colmar, PA 18915, 215-822-7727)

$1431 for 25 cps, from Microhouse
(Box 498F, Bethlehem, PA 18016, 215-868-8219)

$1375 for 25 cps, $1885 for 45 cps, from B G Sound
(60 E. 9th St., New York, NY 10003, 212-674-1185)

DAISY WHEEL 2
about $1700

Another low-cost daisy-wheel printer is the one manufactured by Ricoh (another Japanese company). Radio Shack calls it the "Daisy Wheel 2" and lists it at $1960. You can get it from discount dealers:

$1695 from Computer Discount of America
(W. Milford Mall, W. Milford, NJ 07480, 201-728-8080, 800-526-5313)

$1666 from Rand's
(2185 East Fry Blvd., Sierra Vista, AZ 85635, 602-458-2477)

Its speed is 43 cps. Though it costs more than the 25 cps version of the Starwriter, it costs less than the 45 cps version of the Starwriter. It attaches easily to Radio Shack's models 1, 2, and 3. If you want to attach it to an Apple, you must alter one of the wires in its cable.

ELECTRONIC 50
about $2000

For the highest quality printing, get IBM's Electronic Typewriter model 50 (which costs $1430), and attach it to your computer by using Mediamix's interface (which costs $495). The "Electronic 50" prints more beautifully than any other kind of printer. It's what I use to produce most of these books.

The Electronic 50 is a fancy typewriter that contains a tiny IBM computer, which lets it do some word-processing. To do more elaborate word-processing, you must attach it to a larger computer (such as Radio Shack's model 3), by using Mediamix's interface.

Though Mediamix's interface normally costs $495, I recommend that you buy the souped-up version, which costs $570 instead. The souped-up version makes the right-hand margin perfectly straight. (To buy the souped-up version, ask for the interface "with the JPS Option".)

I haven't bought Mediamix's interface yet, which is why my right-hand margin isn't straight.

The Electronic 50 prints slowly: only 15 cps. But if you care about beautiful printing more than about speed, the Electronic 50 is what you want.

Buy the Electronic 50 from your local branch of IBM. Buy the interface from Mediamix (Box 67B57, Los Angeles, CA 90067, 213-475-9949). The head of Mediamix, Tim Wilde, knows a lot about typesetting. IBM approves of his interface— and so do I.

SPINWRITER
about $2400

The Spinwriter is made by a Japanese company: Nippon Electric Company, whose American subsidiary is called "NEC", which is pronounced "neck". The Spinwriter is nice, reliable, sturdy, letter-quality. It uses a thimble, which is better than a daisy wheel. It prints almost as nicely as a typewriter, and has the advantage of going much faster.

It comes in several models. You'll probably want the "5510", which goes at 55 cps and lacks a keyboard. You can get a fancier model that has a keyboard. NEC is coming out with a new, cheaper model, that goes at only 35 cps.

You can get the 55 cps model, without keyboard, for just $2395 from The CPU Shop (5 Dexter Row, Charlestown, MA 02129, 617-242-3361, 800-343-6522).

OBITUARIES

Because of recent advances in printer technology, Radio Shack has discontinued the Line Printer 1 (Centronics 779), the Line Printer 3, the Quick Printer 1 (Centronics P1), the Daisy Wheel 1, and the Screen Printer.

Number systems

COMPUTERS AREN'T HUMAN

Most humans use the *decimal system*, which consists of ten digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), because humans have ten fingers. The computer does not have fingers, so it prefers other number systems instead. Here they are....

BINARY

Look at these powers of 2:

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$

Now try an experiment: pick your favorite positive integer, and try to write it as a sum of powers of 2. For example, suppose you pick 45. You can write it as 32+8+4+1. Suppose you pick 74; you can write it as 64+8+2. Suppose you pick 77; you can write it as 64+8+4+1. *Every* positive integer can be written as a sum of powers of 2.

Let's put those examples in a table:

ORIGINAL NUMBER	WRITTEN AS A SUM OF POWERS OF 2	DOES THE SUM CONTAIN... 64?	32?	16?	8?	4?	2?	1?
45	32+8+4+1	no	yes	no	yes	yes	no	yes
74	64+8+2	yes	no	no	yes	no	yes	no
77	64+8+4+1	yes	no	no	yes	yes	no	yes

To write those numbers in the *binary system*, replace "no" by 0 and "yes" by 1:

DECIMAL SYSTEM	BINARY SYSTEM
45	0101101 (or simply 101101)
74	1001010
77	1001101

The *decimal system* uses the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 and uses these columns:

thousands hundreds tens units

For example, the decimal number 7105 means "7 thousands + 1 hundred + 0 tens + 5 units".

The *binary system* uses only the digits 0 and 1, and uses these columns:

sixty-fours thirty-twos sixteens eights fours twos units

For example, the binary number 1001101 means "1 sixty-four + 0 thirty-twos + 0 sixteens + 1 eight + 1 four + 0 twos + 1 unit"; in other words, it means seventy-seven.

In elementary school, you were taught how to do arithmetic in the decimal system. You had to memorize the addition and multiplication tables:

DECIMAL ADDITION

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

DECIMAL MULTIPLICATION

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	10	12	14	16	18
3	0	3	6	9	12	15	18	21	24	27
4	0	4	8	12	16	20	24	28	32	36
5	0	5	10	15	20	25	30	35	40	45
6	0	6	12	18	24	30	36	42	48	54
7	0	7	14	21	28	35	42	49	56	63
8	0	8	16	24	32	40	48	56	64	72
9	0	9	18	27	36	45	54	63	72	81

In the binary system, the only digits are 0 and 1, so the tables are briefer:

BINARY ADDITION

	0	1
0	0	1
1	1	10

← *because two is written "10" in binary*

BINARY MULTIPLICATION

	0	1
0	0	0
1	0	1

If society had adopted the binary system instead of the decimal system, you would have been spared many hours of memorization!

Usually, when you ask the computer to perform a computation, it converts your numbers from the decimal system to the binary system, performs the computation by using the binary addition and multiplication tables, and then converts the answer from the binary system to the decimal system, so you can read it. For example, if you ask the computer to print 45+74, it will do this:

```
 45 converted to binary is   101101
+74 converted to binary is +1001010
                            1110111   converted to decimal is 119
```

because 1+1=10

The conversion from decimal to binary and then back to decimal is slow. But the computation itself (in this case, addition) is quick, since the binary addition table is so simple. The only times the computer must convert is during input (decimal to binary) and output (binary to decimal). The rest of the execution is performed quickly, entirely in binary.

You know fractions can be written in the decimal system, by using these columns:

units point tenths hundredths thousandths

For example, $1\frac{5}{8}$ can be written as 1.625, which means "1 unit + 6 tenths + 2 hundredths + 5 thousandths".

To write fractions in the binary system, use these columns instead:

units point halves fourths eighths

For example, $1\frac{5}{8}$ is written in binary as 1.101, which means "1 unit + 1 half + 0 fourths + 1 eighth".

You know $\frac{1}{3}$ is written in the decimal system as 0.3333333..., which unfortunately never terminates. In the binary system, the situation is no better: $\frac{1}{3}$ is written as 0.010101.... Since the computer stores only a finite number of digits, it cannot store $\frac{1}{3}$ accurately—it stores only an approximation.

A more distressing example is $\frac{1}{5}$. In the decimal system, it's .2, but in the binary system it's .0011001100110011.... So the computer can't handle $\frac{1}{5}$ accurately, even though a human can.

Suppose you run this BASIC program on a PDP-10 computer:

```
10 PRINT "MY FAVORITE NUMBER IS"4.001-4
20 END
```

The computer will try to convert 4.001 to binary. Unfortunately, it can't be converted exactly; the computer's binary approximation of it is slightly too small. The computer's final answer to 4.001-4 is therefore slightly less than the correct answer. Instead of printing MY FAVORITE NUMBER IS .001, the computer will print MY FAVORITE NUMBER IS .000999987.

If your computer isn't a PDP-10, it's approximation will be slightly different. To test your computer's accuracy, try 4.0001-4, and 4.00001-4, and 4.000001-4, etc. You might be surprised at its answers.

Let's see how the PDP-10 handles this:

```
10 FOR X = 7 TO 193 STEP .1
20     PRINT X
30 NEXT X
40 END
```

The computer will convert 7 and 193 to binary accurately, but will convert .1 to binary only approximately; the approximation is slightly too large. The last few numbers it should print are 192.8, 192.9, and 193, but because of the approximation it will print slightly more than 192.8, then slightly more than 192.9, and then stop (since it is not allowed to print anything over 193).

There are only two binary digits: 0 and 1. A *b*inary dig*it* is called a *bit*. For example, .001100110011 is a binary approximation of $\frac{1}{5}$ that consists of twelve bits. A sixteen-bit approximation of $\frac{1}{5}$ would be .0011001100110011. A bit that is 1 is called *turned on*; a bit that is 0 is *turned off*. For example, in the expression 11001, three bits are turned on and two are off. We also say that three of the bits are *set* and two are *cleared*.

All information inside the computer is coded, in the form of bits:

PART OF THE COMPUTER	WHAT A 1 BIT IS	WHAT A 0 BIT IS
electric wire	high voltage	low voltage
punched paper tape	a hole in the tape	no hole in the tape
punched IBM card	a hole in the card	no hole in the card
magnetic drum	a magnetized area	a non-magnetized area
core memory	a core magnetized clockwise	a core magnetized counterclockwise
flashing light	the light is on	the light is off

For example, to represent 11 on part of a punched paper tape, the computer punches two holes close together. To represent 1101, the computer punches two holes close together, and then another hole farther away.

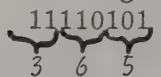
OCTAL

Octal is a short-hand notation for binary:

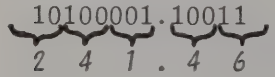
OCTAL	MEANING
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Each octal digit stands for three bits. For example, the octal number 72 is short for 111010.

To convert a binary integer to octal, divide the number into chunks of three bits, starting at the right. For example, here's how to convert 11110101 to octal:



To convert a binary real number to octal, divide the number into chunks of three bits, starting at the decimal point and working in both directions:

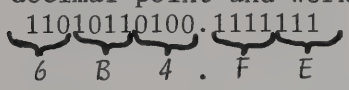


HEXADECIMAL

Hexadecimal is another short-hand notation for binary:

HEXADECIMAL	MEANING
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

For example, the hexadecimal number 4F is short for 01001111. To convert a binary number to hexadecimal, divide the number into chunks of 4 bits, starting at the decimal point and working in both directions:



Character codes

ASCII

To store a character in a string, the computer uses a code. The most famous code is the *American Standard Code for Information Interchange (ASCII)*, which has 7 bits for each character. Here are examples:

CHARACTER	ASCII CODE	ASCII CODE IN OCTAL
space	0100000	040
!	0100001	041
"	0100010	042
#	0100011	043
$	0100100	044
%	0100101	045
&	0100110	046
'	0100111	047
(	0101000	050
)	0101001	051
*	0101010	052
+	0101011	053
,	0101100	054
-	0101101	055
.	0101110	056
/	0101111	057

CHARACTER	ASCII CODE	ASCII CODE IN OCTAL
0	0110000	060
1	0110001	061
2	0110010	062
etc.		
9	0111001	071
:	0111010	072
;	0111011	073
<	0111100	074
=	0111101	075
>	0111110	076
?	0111111	077
@	1000000	100
A	1000001	101
B	1000010	102
C	1000011	103
etc.		
Z	1011010	132
[	1011011	133
\	1011100	134
]	1011101	135
↑	1011110	136
←	1011111	137

"ASCII" is pronounced "ass key".

PDP-10 computers use mainly 7-bit ASCII, but can also handle a "6-bit ASCII", formed by eliminating ASCII's second bit: for example, the 6-bit ASCII code for the symbol $ is 0 00100. PDP-8 computers use mainly a "6-bit ASCII" formed by eliminating 7-bit ASCII's leftmost bit, but can also handle an "8-bit ASCII" formed by putting a 1 before 7-bit ASCII. The PDP-11 and most microcomputers use an "8-bit ASCII" formed by putting a 0 before 7-bit ASCII. Most terminals use 7-bit ASCII. CDC computers use a special CDC 6-bit code.

EBCDIC

Instead of using ASCII, IBM computers use the *Extended Binary-Coded-Decimal Interchange Code (EBCDIC)*, which has 8 bits for each character. Here are examples:

CHARACTER	EBCDIC CODE IN HEXADECIMAL	CHARACTER	EBCDIC CODE IN HEXADECIMAL	CHARACTER	EBCDIC CODE IN HEXADECIMAL
space	40	-	60	A	C1
¢	4A	/	61	B	C2
<	4C	,	6B	etc.	
(	4D	%	6C	I	C9
+	4E	_	6D	J	D1
|	4F	>	6E	K	D2
&	50	?	6F	etc.	
!	5A	:	7A	R	D9
$	5B	#	7B	S	E2
*	5C	@	7C	T	E3
)	5D	'	7D	etc.	
;	5E	=	7E	Z	E9
¬	5F	"	7F	0	F0
				1	F1
				etc.	
				9	F9

"EBCDIC" is pronounced "ebb sih Dick".

IBM 360 computers can also handle an "8-bit ASCII", formed by copying ASCII's first bit after the second bit: for example, the 8-bit ASCII code for the symbol $ is 01000100. But IBM 370 computers (which are newer than IBM 360 computers) don't bother with ASCII: they stick strictly with EBCDIC.

80-column IBM cards use *Hollerith code*, which resembles EBCDIC but has 12 bits instead of 8. 96-column IBM cards use a 6-bit code that's an abridgement of the Hollerith code.

Here's a program in BASIC:

```
10 IF "A"<"9" THEN 100
20 PRINT "CAT"
30 STOP
100 PRINT "DOG"
110 END
```

Which will the computer print: CAT or DOG? The answer depends on whether the computer uses EBCDIC or ASCII.

Suppose the computer uses EBCDIC. Then the code for "A" is hexadecimal C1, and the code for "9" is hexadecimal F9. Since C1 is less than F9, the computer considers "A" to be less than "9", so the computer prints DOG.

But if the computer uses 7-bit ASCII instead of EBCDIC, the code for "A" is octal 101, and the code for "9" is octal 071; since 101 is greater than 071, the computer considers "A" to be greater than "9", so the computer prints CAT.

BYTES

A *byte* usually means: eight bits. For example, here's a byte: 10001011.

For computers that use 7-bit ASCII, programmers sometimes define a byte to be 7 bits instead of 8. For computers that use 6-bit ASCII, programmers sometimes define a byte to be 6 bits. So if someone tries to sell you a computer whose memory can hold "16,000 bytes", he probably means 16,000 8-bit bytes, but might mean 7-bit bytes or 6-bit bytes.

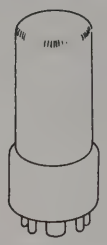
NIBBLES

A *nibble* is 4 bits. It's half of an 8-bit byte. Since a hexadecimal digit stands for 4 bits, a hexadecimal digit stands for a nibble.

Chips

GENERATIONS

In old computers, built before 1959, the CPU consisted of thousands of vacuum tubes. A vacuum tube looked like a cylindrical light bulb:



The tube was made of glass, and had a vacuum inside. Those old computers were called *first-generation*.

In *second-generation* computers, built from 1959 to 1963, the CPU consisted of thousands of discrete transistors. A discrete transistor was a metal cylinder with three wires sticking out:

Compared with vacuum tubes, discrete transistors were smaller, cheaper, more reliable, and consumed fewer watts of electricity.

In *third-generation* computers, built since 1964, the CPU consists of *integrated circuits*. An *integrated circuit (IC)* is a tiny slab of metal. A single IC does as much work as a whole pile of transistors; so by using integrated circuits, computers have become even smaller, cheaper, more reliable, and consume less electricity.

If the integrated circuit does the work of fewer than 10 transistors, it's called *small-scale integrated (SSI)*; if it does the work of 10 to 99 transistors, it's *medium-scale integrated (MSI)*; if it does the work of at least 100 transistors, it's *large-scale integrated (LSI)*; if it does the work of at least 10,000 transistors, it's *very-large-scale integrated (VLSI)*. VLSI is better than LSI, which is better than MSI, which is better than SSI. LSI began in 1969; computers using LSI (or VLSI) are usually called *late third-generation computers*, though salesmen sometimes call them *fourth-generation* computers.

An integrated circuit is also called a *semiconductor chip*— because it's made of semiconductive silicon. Most are manufactured in a part of California that's been nicknamed "Silicon Valley".

If the chip is a complete CPU, it's called a *microprocessor*. If you buy a Radio Shack or an Apple or any other popular microcomputer, the CPU is all on one chip and is therefore a microprocessor.

In every modern computer, the CPU consists of chips (one or many). Other chips make up the RAM and ROM. Some chips act as *interfaces*, which help the various parts of the computer communicate with each other.

DIPS

When you buy a chip, it comes already mounted in the belly of a "caterpillar", which is called a *dual in-line package (DIP)*. The caterpillar's legs are called *pins*. Usually, if the chip is a CPU (a microprocessor), the DIP is 2" long and has 40 pins; if the chip is a ROM, the DIP is smaller ($1\frac{1}{4}$") and has 24 pins; if the chip is a RAM, it contains 16,384 bits and has a DIP that's even smaller ($\frac{3}{4}$" with 16 pins).

PRICES

If you contact a chip manufacturer and ask to buy a chip, the manufacturer will give it to you in a DIP, at no extra charge. So "the price of the chip" actually means "the price of the chip and its DIP".

Here's how much you have to pay, to buy a microprocessor:

Microprocessor	Price for chip (& DIP)	Computers using that chip (or a variant)
Zilog Z-80	$8.45	Radio Shack (models 1 & 3), some CP/M
Zilog Z-80A	$9.45	Radio Shack (model 2), Sinclair, most CP/M
MOS Technology 6502	$7.85	Apple, Atari, VIC, PET, CBM, Ohio Sci.
Intel 8080A	$4.95	Compucolor, Intecolor
Motorola 6800	$5.45	SWTP
Intel 8085	$8.98	some CP/M

Those prices are from Active Electronic Sales (Box 1035, Framingham, MA 01701, 617-879-0077; your order must total at least $10; add $3 to your order, for shipping).

How much does a RAM chip cost? The answer depends on which kind of RAM chip you want. The most popular kind is the one invented by Texas Instruments, called the "model 4116"; it contains 16,384 bits. If you buy eight of them, you get a total of 16,384 *bytes* (since a byte is eight bits). You can buy eight of them for a total price of just $19 (plus shipping) from The CPU Shop (5 Dexter Row, Charlestown, MA 02129, 617-242-3361, 800-343-6522). So for just $19, you can get 16K of RAM.

BIPOLAR VERSUS MOS

A chip can be made in two ways. The old way is called *bipolar*; the new way is called *metal-oxide semiconductor field-effect transistor (MOSFET*, or simply *MOS*, which is pronounced "moss"*)*.

Each way has its own advantages. Bipolar chips have the advantage of being faster. MOS chips, though slower, are cheaper, consume less power, and are denser (contain more intelligence and memory per square inch). Because of their low cost, MOS chips are more popular than bipolar. Microcomputers use only MOS: bipolar is too expensive. Maxicomputers use MOS for most of the memory, but use bipolar for the most important parts of the memory and for the CPU.

A MOS chip has many electrical *channels* in it. In *p-channel MOS (PMOS)*, all the channels are positively charged: they contain silicon that has been "doped" with positively charged boron. In *n-channel MOS (NMOS)*, all the channels are negatively charged: they contain silicon that has been doped by negatively charged phosphorus. In *complementary MOS (CMOS)*, the even-numbered channels are negative, and the odd-numbered channels are positive.

NMOS is faster than PMOS. Of all the kinds of MOS, CMOS is the fastest and consumes the least power; but it's also the most expensive and the least dense. Because of its high price, computers use it less often than PMOS and NMOS. But every digital wristwatch uses CMOS instead of PMOS or NMOS, because the watch needs a chip that consumes little power, so it can run for many days on a tiny battery.

All the popular microcomputers use just NMOS. For example, the model 4116 RAM, the Z-80 CPU, and the 6502 CPU are all NMOS. But a few "weird" microcomputers use other kinds of MOS instead. Texas Instruments computers (such as the TI-99/4 and the TI-990) use the TI-9900 CPU, which is PMOS, which is why Texas Instruments computers run slowly. RCA computers (such as the COSMAC, the VIP, and the Elf) use the RCA 1802 CPU, which is CMOS, which is why RCA computers require very little electricity. Another CMOS CPU is the Intersil 6100, which imitates a PDP-8.

KINDS OF RAM

A RAM chip is either *dynamic* or *static*. If the chip is dynamic, the data stored in it lasts only 2 milliseconds; after the 2 milliseconds, the electrical charges representing the bits dissipate, and become too weak to detect. If you want to store the data longer, you must read that data before 2 milliseconds go by, then write it onto the chip again, then read it again before 2 milliseconds go by, etc. In other words, every 2 milliseconds you must *refresh* the data, by reading it and rewriting it. When you buy a *dynamic RAM board* (a green printed-circuit board that has dynamic RAM DIPs mounted on it), the board also contains a *refresh circuit*, which refreshes the data automatically and continually.

If the chip is static instead of dynamic, the electrical charge never dissipates, so you don't need a refresh circuit. (However, you must still keep the power turned on.)

In the past, computer designers were afraid of dynamic RAM, and stuck with static; but dynamic RAM has finally become more popular, because it is cheaper than static, and because manufacturers have invented circuits that refresh the RAM without slowing down the CPU.

Today, the most popular kind of RAM is dynamic NMOS. (For example, the "4116" is dynamic NMOS.) Static NMOS is still available. CMOS and bipolar are always static.

KINDS OF ROM

An ordinary ROM (read-only memory) contains information the computer can read but can't change. The information in the ROM is called *firmware*.

When you buy a ROM, it usually has the information stored in it already. But if you prefer, you can buy a *programmable ROM (PROM)*, which is blank and lets you program your own information into it. To put information into the PROM, you must buy a device called a *PROM programmer*. Information put into the PROM cannot be changed— unless the PROM is *erasable*. An erasable PROM is called an *EPROM*.

You can buy two kinds of EPROM. The usual kind is called an *ultraviolet-erasable PROM (UV-EPROM)*: you erase it by shining an intense ultraviolet light at it for 20 minutes. The more expensive kind of EPROM is called an *electrically erasable PROM (EEPROM)* or *electrically alterable PROM (EAPROM)*: you erase it by sending a 25-volt current through it for a tenth of a second.

After an EPROM is erased, you can put new information into it.

Scheduling the users

CPU

The first computers were invented about 1950. Back then, the only kind of computer you could buy was a *single-user system*: only one person could use the computer. If two people wanted to use the computer, the second person had to stand in line behind the first person, until the first person finished.

Batch

The first improvement to that system was *batch*. In a computer that had *batch processing*, the second person didn't have to stand in line to use the computer; instead, the second person could feed his program into the computer also, even though the first person wasn't done. The second person's program waited in a special part of the computer's memory (RAM or core or disk or drum or tape), until the first person's program was done; then the computer would start working on the second program.

It was called a *batch* system, because the people could insert a batch of two programs, instead of inserting just one program at a time. In most batch systems, the people could insert a batch of *many* programs: the computer would do each one in order.

Multiprogramming

A computer system consists of many parts: the CPU, the printer, the disk (or tape), etc. Some parts of the system work more slowly than others.

For example, the printer is slower than the CPU. If the CPU tells the printer to print some information, the CPU must wait until the printer finishes.

While the CPU is waiting, why not let the CPU start working on the second person's program? If the CPU is smart enough to do that, it's called a *multiprogramming system*.

So in a simple multiprogramming system, the CPU uses this strategy: the CPU begins by working on the first person's program; but if work on that program must be temporarily interrupted (because of waiting for the printer, or some other slow device), the CPU starts working on the second program. (Similarly, if the second program must be temporarily interrupted also, the CPU starts working on a third program.) But the first program always has top priority: as soon as the first program can continue (because the printer has finished obeying the CPU's command), work on the first program resumes, and all the other programs are put on hold again.

A difficulty

That simple multiprogramming system is quite practical and efficient— except for the following difficulty....

Suppose two guys walk up to the computer. One guy's program is very long (it figures out the entire payroll for a large company); the other guy's program is short (it just adds 2+2). If the guy with the longer program is kind, he'll let the other guy go first. But what if the guy with the long program is mean and shoves his way to the computer first? Then the poor second guy (who wanted to just add 2+2) will have to wait until the entire payroll finishes (unless the first program uses the printer, which makes the CPU switch attention to the second program).

It would be nice if the CPU could "psyche out" the situation, and give attention to the second guy, without waiting for the first guy to finish. That way, the second guy could leave in a few seconds, instead of having to wait many hours.

Time-slicing

The easiest way to make the CPU get to the second guy quickly is to use a technique called *round-robin time-slicing*. On most large computers, the "time-slice" is a sixtieth of a second. Here's how it works....

The CPU works on the first guy's program for a sixtieth of a second. Then the CPU switches its attention to the second guy, and gives him a sixtieth of a second. Then the CPU switches to the third guy, and gives him a sixtieth of a second. And so on, until everybody's program has received a sixtieth of a second. Then the CPU returns to the first program, and gives it another sixtieth of a second. Then— to the second program. And so on. If somebody's program is short, he'll be done fairly soon, and won't have to wait several hours for the long-winded guys ahead of him to finish.

If the computer uses that round-robin time-slicing, each person has his own terminal. The many terminals are attached to the computer. The CPU goes round and round, switching its attention from terminal to terminal every sixtieth of a second.

It all happens so fast that each person is under the illusion that the CPU is thinking about him the whole time. But really, the CPU is paying attention to just one person at a time, and switches its attention from person to person very fast. That's called *timesharing*. It's where the CPU juggles several programs so quickly that each person is under the *illusion* that the CPU is thinking about just him the whole time.

In that system, a person might not get a full sixtieth of a second on his turn. For example, if the person's program needs to use the printer, the CPU sends some data out to the printer but then immediately moves on to the next person, without waiting for the printer to catch up. When the CPU has given all the other people their time slices and returns again to the person who wanted the printer, the CPU checks whether the printer has finished the job yet.

Fewer switches

Unfortunately, "switching attention from one person to another" is complicated: the CPU needs several milliseconds to make the switch. It would be nice if the CPU could switch less often, yet still get to each person fast enough so that each person still thinks the CPU is thinking only about him.

For example, suppose a person is a "CPU hog": he's running a program that won't finish for several hours. Instead of giving that person many little time-slices, the CPU could act more efficiently by by totally ignoring that person for several hours (which will make everybody else in the computer room quite happy!), and then give that person a solid block of time toward the end of those hours. He'll never know the difference: his job will finish at the same time as it would have otherwise. And the CPU will have an easier time, since it doesn't have to switch its attention to and from him so many times.

But how can the CPU decide which person is the "hog"? Make the CPU count how many time-slices and how much RAM each person has already been using; if a person's count is high, the person has been acting hoggish. The CPU assumes that the person will probably continue to be a hog; so that person is ignored until later, when he's given a solid block of time.

But if the "solid block of time" is *too* long, all the other people will be ignored for too long and will think the computer broke. So the "solid block of time" shouldn't be more than a few seconds. Nevertheless, "a few seconds" is a lot longer than a *sixtieth* of a second! If the program still isn't done after the solid block of time, give the program *another* solid block of time, later.

The PDP-20 and other top-notch timesharing systems use those strategies.

Contrasts

Remember: a computer that has "batch" might not be fancy enough to have "multiprogramming"; and a computer that has "multiprogramming" might not be fancy enough to have "timesharing".

During the 1970's, a popular question was, "Does your computer use timesharing, or just batch?"

Most timesharing systems use terminals with screens; most just-batch systems use punched cards instead. Timesharing (using terminals that have screens) is becoming more popular than just batch (and just punched card).

Distributed intelligence

The whole point of that discussion was to show how to make many people share a single CPU efficiently. But why bother? The CPU chip inside a Radio Shack or Apple computer costs less than $10; why bother sharing it? Why not simply give each person his own CPU?

So today many people are getting rid of maxicomputers that have fancy timesharing operating systems, and are replacing them by a collection of microcomputers, each of which handles only one person at a time.

That's called *distributed intelligence*: tying together many little CPU's, instead of forcing everybody to share one big CPU.

MEMORY

Suppose you want to run a program that's too long to fit in the RAM. What should you do? The obvious answer: chop the program into little pieces, and run one piece at a time.

Virtual memory

If you buy a computer that has *virtual memory*, the computer will chop the program automatically. The operating system will automatically chop your program into little pieces, put as many pieces as possible into the RAM, and run them. The remaining pieces are put onto a disk instead. When the CPU finishes processing the pieces in the RAM, and needs to get to the pieces on the disk, the operating system automatically fetches those pieces from the disk and copies them into the RAM— after copying the no-longer-needed pieces from the RAM to the disk.

Each piece of the program is called a *page*. On most computers, each page is 4K. Copying a page from the disk to the RAM (or vice versa) is called *paging* or *swapping*.

If many people try to share the RAM, the operating system tries to fit all their programs into the RAM simultaneously. But if they won't all fit, the operating system automatically puts some of their programs onto the disk temporarily, until the other programs are done (or have used up a time-slice).

Thrashing

If too many people try to use the computer at once, and the computer's RAM is too small, the operating system will spend lots of time moving programs (or pages) to and from the disk. That's bad. In that situation, we say the operating system is *thrashing*. It's spending so much time moving between the RAM and the disk that hardly any time remains for the CPU to run the programs! The computer system is spending more time on bureaucracy than on doing useful work! The solution: buy more RAM, or tell some of the people to go home!

I/O DEVICES

I/O devices (such as printers and terminals) are much slower than the CPU. After the CPU sends a command to a slow device, the CPU must wait for the device to obey. While the CPU is waiting, the CPU ought to be doing something useful— such as processing somebody else's program.

Buffers

Here's how the CPU handles the problem of "waiting". When the CPU wants to send a message (or command) to a slow device, it puts the message into a *buffer*, which is a special part of the RAM. The device peeks at the buffer, and obeys what the message says. While the device is handling the message in the buffer, the CPU's attention switches to a different program— or to a different part of the same program.

The buffer is in RAM. But *which* RAM? If you buy a new computer system, each I/O device contains a little RAM (usually 2K). The "little RAM inside the I/O device" is the buffer. That's if your equipment is ultra-modern. If your equipment is old-fashioned, the buffer is instead part of the ordinary RAM (the RAM that's in the same cabinet as the CPU).

Buffers are used not only for printers and terminals but also for disk drives and tape drives. Each slow device needs its own buffer.

Spooling

Here's another nasty problem. Suppose your program says to write answers on the printer, but somebody else has been using the printer and hasn't finished yet.

In that situation, the CPU *pretends* to obey your command; it *pretends* to write your answers on the printer. But actually, it writes your answers on a disk instead. Then, when the previous person finishes using the printer, the CPU copies your answers from the disk to the printer (and erases them from the disk). That all happens automatically, thanks again to the operating system. The part of the operating system that takes care of waiting-for-the-printer is called the *spooler*. Putting your answers temporarily onto a disk is called *spooling*.

Spooling is used mainly for the printer, but can also be used for other output devices, such as the plotter.

The word "spool" is an abbreviation for "simultaneous processing of on-line" devices. If several people what to use the printer simultaneously, the operating system must know how to spool.

If you want to use the printer, and somebody else is in the middle of using the printer to print paychecks, and your operating system is so crummy that it doesn't have a spooler, what happens? Either the operating system tells you "Sorry, but I can't continue your program until the other guy is done— please wait" or the operating system makes a mess— by printing your answers in the middle of the other guy's paycheck!

Print queue

Suppose your operating system *does* have a spooler, so that it prints your answers temporarily onto a disk. What happens if a third person wants to use the printer also? Yep, that person's answers get printed onto the disk also. If the printer is very slow, many people might have their answers waiting on the disk. The answers that are waiting on the disk are called the *printer's waiting-line* or the *printer's queue* or (even more briefly) the *print queue*.

"Queue" is pronounced "kyoo"; it's a British word that means "waiting-line". In British supermarkets, the customers don't wait in lines, they "wait in queues"; they "queue up". So do the answers that are waiting to go to the printer.

Personality

COMPUTERS HAVE PERSONALITIES

A computer can be a boss, a conversationalist, or a drudge.

BOSS

A "boss" is a computer that monitors another machine. The "other machine" might be part of a factory's assembly line, a microwave oven, a burglar-alarm system, a series of traffic lights, the sound system for an electronic organ, or a life-support system that periodically gives intravenous injections to an unconscious patient. The computer makes sure everything is running smoothly. Usually, you won't even know the computer is there: it may be hidden inside the machine it monitors.

Instead of calling such a computer a "boss", computer ads call it a *controller* or a *scientific real-time system*.

CONVERSATIONALIST

A "conversationalist" is a computer you can chat with, by using a terminal or a keyboard-with-display. The conversation might be about a game you're playing against the computer, a reservation for an airplane flight, or your personality (if the computer is trying to play therapist). Throughout the conversation, what you say to the computer is brief, and so are the computer's replies. Typically, either you're asking the computer for more output, or the computer is asking you for more input.

Instead of calling such a computer a "conversationalist", computer ads call it an *interactive computer*. A conversationalist whose memory contains lots of data, and that answers questions about the data, is called a *data-inquiry system* or a *commercial real-time system*. A *timesharing computer* is a conversationalist that handles many programs simultaneously, with each program on a different terminal.

DRUDGE

A "drudge" is a computer that takes in piles of data and spits out piles of answers. Typically, the data is on cards, and the answers are on line-printer paper. While the computer is working, it doesn't ask you any questions; you never need to speak with it. It just does its job faithfully, blue-collar style.

A *batch processor* is a drudge that lets you start feeding in a second program before the first program has finished. If the batch processor starts *working on* the second program before the first program has finished— in other words, if it works on both programs simultaneously— it's called a *multiprogramming batch processor*.

Operating systems

IBM

For the IBM 360 computer, IBM invented an Operating System called *OS*. IBM also invented a very stripped-down version of the operating system, called *Basic OS (BOS)*. If your computer had too little memory for OS, you'd use BOS instead. IBM also invented two intermediate versions, smaller than OS but larger than BOS. The intermediate versions were called the *Tape OS (TOS)* and the *Disk OS (DOS)*.

Nobody uses BOS or TOS anymore. So the only popular operating systems today are OS and DOS. If your computer has lots of memory, it uses OS; if it has too little memory for OS, it uses DOS instead.

To begin using an IBM computer, you must give commands that the operating system understands. The commands it understands are called the *Job Control Language (JCL)*. So before using an IBM computer, you must read a book that explains JCL. The version of JCL depends on which operating system you have: the JCL for OS is fancier than the JCL for DOS.

Your first JCL command tells the computer your name and account number and password, so the computer can decide whether you're an authorized user. Later, you must give a JCL command that tells the computer which language you want to use: BASIC, FORTRAN, PASCAL, COBOL, or a language that's weirder.

The original version of DOS was called simply "DOS". Later, IBM invented a better version of DOS, called *DOS with Virtual Storage (DOS/VS)*. To use DOS/VS, you must buy IBM's newer computer, which is called the IBM 370.

The original version of OS was called *OS with Multiprogramming a Fixed number of Tasks (OS/MFT)*. Then IBM invented a fancier version, called *OS with Multiprogramming a Variable number of Tasks (OS/MVT)*. Those versions of OS work on the IBM 360. For IBM's newer computer— the IBM 370— IBM invented *OS/VS1*, then an even fancier version, called *OS/VS2*.

For flexibility, get the *Virtual Machine operating system (VM)*, which is an operating system that lets you invent your own operating system, and lets you run five different operating systems simultaneously! If you use VM, you'll probably make one of your five operating systems be OS/VS1 (or OS/VS2), and make another one be *CMS* (which stands for *Cambridge Monitor System* or *Conversational Monitor System*).

DEC

DEC's largest computer is called the PDP-20 or Decsystem-20. Its operating system is called *The OPerating System 20 (TOPS-20)*. DEC's previous large computer was called the PDP-10 or Decsystem-10; its operating system was called *TOPS-10*. The commands that the operating system understands are called *monitor commands*. So if your computer is a PDP-20 or PDP-10, you must learn DEC's monitor commands (instead of learning IBM's JCL).

DEC also makes a popular minicomputer, called the PDP-11. Several operating systems are available for the PDP-11. The most popular operating system is called the *Resource-Sharing Timesharing System Extended (RSTS/E)*. It can handle many terminals, and is intended for programming in BASIC-PLUS or BASIC-PLUS-2. If your PDP-11 is too small to handle RSTS/E, get DEC's smaller operating system, which is called *Real-Time 11 (RT-11)*; it handles fewer terminals and is intended for FORTRAN or a stripped-down version of BASIC (called BASIC-11, which is worse than BASIC-PLUS or BASIC-PLUS-2). If your PDP-11 is large, and you want it to handle a large program in COBOL (instead of many small programs in BASIC), get the *Resource Sharing eXtended 11 operating system (RSX-11)* or a souped-up version of it, called the *Interactive Application System (IAS)*. If you're programming in a weird language called MUMPS, get the *Data System Manager 11 (DSM-11)* instead.

DEC's newest computer is a very fancy minicomputer (fancier than the PDP-11) and is called the Virtual Address eXtended 11 (VAX-11). Its operating system is called the *VAX Virtual Memory System (VAX/VMS)*. To give a command to VAX/VMS, you use *DEC Control Language (DCL)*— instead of IBM's JCL.

RADIO SHACK

Radio Shack's first computer was called the TRS-80 model 1. Its ROM contained BASIC and the fundamental operating system, which controlled the screen and keyboard and cassette. But the ROM did *not* contain the part of the operating system that handles the disk. If you bought a disk drive, Radio Shack gave you a disk containing the "disk part of the operating system". That part was called the *TRS Disk Operating System (TRSDOS)*. The first version of TRSDOS that Radio Shack distributed was called version 2.1; it contained a few errors. Later, Radio Shack distributed a better version, called 2.2, which contained fewer errors. Finally, Radio Shack distributed version 2.3, which was perfect.

Another company, called Apparat, invented a different disk operating system for the TRS-80 model 1. Apparat's DOS was called *Newdos*. It was better than Radio Shack's DOS, but cost more money. Many people bought Newdos. (Newdos users get together to form sex clubs, which are called "Newdos colonies".) Later, Apparat invented a further improvement, called *Newdos/80*.

When Apparat invented Newdos, Apparat bragged that Newdos "corrected the errors in TRSDOS version 2.1". But later, when Radio Shack invented version 2.3, Apparat couldn't brag that Newdos contained fewer errors than TRSDOS. Nevertheless, Newdos is still better than TRSDOS, because Newdos contains more commands than TRSDOS. In Newdos, you can tell the disk drive to use more than 35 tracks; in Newdos, you can tell the printer to print a copy of what's on your screen; in Newdos, you can tell the computer to find the bottom line of your program and list it. You can't do those things in TRSDOS— not heven in TRSDOS version 2.3.

Since Newdos is so similar to TRSDOS, Radio Shack could sue Apparat for copyright violation. To avoid such a lawsuit, the Newdos manual is purposely incomplete, and says that you must buy TRSDOS and TRSDOS's manual before you try to use Newdos.

Radio Shack's newer computers— the model 2 and model 3— have fancier versions of TRSDOS than the model 1. So on the model 2 and model 3, you won't want to spend money on Newdos.

APPLE

For many years, Apple's disk operating system was called *Apple DOS version 3.2*. It organized the disk into 13 sectors.

Recently, Apple Computer Company has invented a better version, called *Apple DOS version 3.3*. It organizes the disk into 16 sectors. It lets you put more information onto the disk than version 3.2.

If you buy an Apple today, the disk controller assumes the disk has 16 sectors. If you try to feed your new Apple (16 sectors) an old disk (13 sectors), your Apple will vomit the disk, unless you feed the Apple beforehand a program called MUFFIN or a program called BASICS, whcih act like Alka-Seltzer.

CP/M

Many microcomputers that have disks use an operating system called *Control Program for Microcomputers (CP/M)*. The fanciest business programs from microcomputers assume you have CP/M. To get CP/M, you buy a disk that contains it. The disk costs about $150. The disk works only if your computer contains a Z-80 CPU (or its predecessor, the 8080 CPU). Most microcomputers contain a Z-80 CPU, but the Apple does not: the Apple contains a 6502 CPU instead. To use CP/M on an Apple, you must put a Z-80 CPU inside your Apple, by using Microsoft's Z-80 Softcard, which is a card that contains a Z-80 CPU and fits in your Apple.

CP/M was invented by a company called Digital Research, which has invented an even fancier version, called *Multi-Programming for Microcomputers (MP/M)*. MP/M can handle up to four terminals easily; CP/M handles only one.

Jobs

PROGRAMMER

A *programmer* is a teacher: the programmer teaches the computer new tricks. For example, the programmer might teach the computer how to do the payroll. To do that, the programmer feeds the computer a list of instructions, that explain tot he computer how to do the payroll. The list of instructions is called a *program*.

Languages

The program is written by using the very limited vocabulary that the computer understands already. In volumes 1 and 2, I explained a vocabulary called BASIC, which consists of words such as PRINT, INPUT, GO TO, IF, THEN, and STOP. That vocabulary— BASIC— is called a *computer language*. It's a small part of English. No computer understands the whole English language. The programmer's job is to translate an English sentence (such as "do the payroll") into language the computer understands (such as BASIC). So the programmer is a translator.

Some computers understand BASIC. Other computers understand a different vocabulary, called COBOL. For example, COBOL uses the words DISPLAY and WRITE instead of PRINT.

Before programming a computer, you must find out which language the computer understands. Does it understand BASIC? Or does it understand COBOL instead? Or does it understand a yet different language? The most popular languages are BASIC, COBOL, FORTRAN, and PASCAL; but there are also *thousands* of others. Your computer understands at least one of those languages; if you're lucky, your computer understands *several* of those languages.

When you apply for a programming job, the first question to ask the interviewer is: which languages does the company's computer understand? Or better yet, ask, "Which language do you want me to program in?" The interviewer will say "BASIC" or "COBOL" or some similar answer and then ask you, "Do you know that language?"

Most microcomputers use BASIC. Most maxicomputers use COBOL for applications to business, and use FORTRAN for applications to engineering. Minicomputers often use BASIC, but sometimes use COBOL or FORTRAN. The language PASCAL, which is newer, isn't used as much as BASIC, COBOL, or FORTRAN yet, but its popular is gradually increasing.

Of the four popular languages, the easiest is BASIC. It's also the most fun. To become a programmer, begin by studying BASIC, then move on to the other languages, which are yukkier. That's why volumes 1 and 2 explain BASIC, and the other languages are withheld until volumes 7 and 8.

Because BASIC is so easy, saying you know "BASIC" is less prestigious than saying you know "FORTRAN" or "COBOL" or "PASCAL". To get lots of prestige, learn *many* languages. Even if the job you're applying for requires only one language, you should say that you know many languages well, to convince the interviewer that you're brilliant.

The most prestigious languages to know are "assembly language" and "machine language", because they're the most difficult. If you can convince the interviewer that you know assembly language and machine language, the interviewer will assume you're God and offer you a very high salary, even if the job doesn't require a knowledge of those languages.

Specific computers

Before going to the interview, learn about the specific computer the company uses. For example, if the company's computer is an IBM 370, study the IBM 370's details. Study its operating system and its languages. If the job requires COBOL, study the particular dialect of COBOL used on the IBM 370. Each computer has its own dialect of COBOL, its own dialect of BASIC, its own dialect of FORTRAN, its own dialect of PASCAL. Usually, the differences between dialects are small, but you must know them. For "assembly language" and "machine language", the differences between dialects are much greater: the "assembly language" or a Radio Shack computer is almost entirely different from the "assembly language" on an IBM 370.

Analysis versus coding

The act of programming consists of two stages. The first stage is to analyze the problem, to make the problem more specific. For example, suppose the problem is, "Program the computer to do the payroll". The first stage is to decide exactly how the company wants the payroll to be done. Should it be done weekly, bi-weekly, semi-monthly, or monthly? Should the records be stored on disks, or on tapes instead? (Disks are nicer, but more expensive.) While computing the payroll checks, what other reports do you want the computer to generate? For example, do you want the computer to also print a report about the employees' attendance, and about how much money each department of the company is spending on salaries? Maybe one of the departments is over-budgeting. And what kind of paychecks do you want the computer to *refuse* to print? For example, if somebody in the company tries to make the computer print a paycheck for a ridiculous amount (such as $1,000,000 or ½¢), you want the computer to refuse (and perhaps signal an alarm).

That stage— analyzing a vague problem (such as "do the payroll") to make it more specific— is called *analysis*. A person who analyzes is called an *analyst* or, more prestigiously, a *systems analyst*. After analyzing the vague problem and transforming it into a series of smaller, more specific tasks, the analyst turns the problem over to a team of *coders*. Each coder takes one of the tasks and translates it into BASIC or COBOL or some other language.

If you're hired to be a "programmer", your first assignment will probably be as a coder. After you gain experience, you'll be promoted to a systems analyst.

The ideal systems analyst knows how to analyze a problem but also has had prior experience as a coder. A systems analyst who knows how to both code and analyze is called a *programmer/analyst*. An analyst who doesn't know how to code— who doesn't know BASIC or COBOL— who merely knows how to break a big problem into a series of little ones— is paid less.

Three kinds of programming

Programming falls into three categories: *development*, *testing*, and *maintenance*. *Development* means inventing a new program. *Testing* means making sure the program works. *Maintenance* means making minor improvements to programs that were written long ago. (The "improvements" consist of eliminating errors that were discovered recently, or making the program conform to changed government regulations, or adding extra features so that the program produces extra reports or handles extra-special cases.) Development is more exciting than testing, which is more exciting than maintenance. So if you're a new programmer, the other programmers will probably "stick you" in the maintenance department. You'll be part of the maintenance crew. Since your job will consist of "cleaning up" old programs, cruel programmers will call you a "computer janitor".

"Application program" versus a "systems program"

Programs fall into two categories. The usual kind of program is called an *application program*; it handles a specific application (such as "payroll" or "chess" or "send rocket to moon"). The other kind of program is called a *systems program*; its only purpose is to help programmers write applications programs.

For example, hidden inside the computer is a program that makes the computer understand BASIC; that program explains to the computer what the words PRINT, INPUT, GO TO, IF, THEN, and STOP mean. That program (which is called the *BASIC language processor*) is an example of a systems program.

Another systems program is called the *operating system*. It tells the computer how to control the disks and printer and terminals. If the operating system is fancy, it even tells the computer how to handle many programmers at once.

Another systems program, found on large computers, is the *editor*. It lets you edit programs written in languages such as COBOL and FORTRAN.

So systems programs are tools, which help programmers write application programs. When you buy a computer, buy some systems programs, so you can create applications programs easily.

A person who invents systems programs is called a *systems programmer*. To become a systems programmer, learn assembly language and machine language.

Creating a systems program is very difficult; so a systems programmer usual gets paid more than an applications programmer.

The word "systems" is prestigious: it's used in the phrase "systems analyst" and in "systems programmer". In some companies, if your boss wants to praise you, the boss will put the word "systems" in front of your title, even if your job has nothing to do with "systems".

How to learn programming

To be a good programmer, you need experience. You can't become a good programmer by just reading books and listening to lectures; you must get your hands on a computer and practice.

If you take a computer course, the books and lectures are much less valuable than the experience of using the school's computer. Spend lots of time in the computer center. Think of the course as just an excuse to get permission to use the school's computer. The quality of the lecturer is less important than the quality of the school's computer center. The ideal computer center uses video terminals instead of punched cards; has a computer that can understand many languages; gives you *unlimited* use of the computer (no "extra charges"); is open 24 hours a day; has enough terminals so you don't have to wait in line for somebody else to finish; has a staff of "teaching assistants" who will answer your questions; has a rack full of easy-to-read manuals that explain how to use the computer; lets you borrow books and manuals, to take home with you; and has *several* kinds of computers, so that you get a broad range of experience. Before you enroll in a computer course, find out whether the school's computer center has those features.

Many computer schools are unnecessarily expensive. To save money, take fewer courses, and buy more books and magazines instead. Better yet, buy a computer yourself, and keep it in your home! You can buy a VIC for just $279; it's a nice computer, excellent for learning BASIC. Keep the VIC for a few months, learn all you can from it, then sell it to a friend for $179— so that using the VIC has cost you just $100 and given you several *months* of education. That's a much better investment of your money that spending many hundreds of dollars for a computer course.

Another cheap way to get an education is to phone your town's board of education, and ask whether the town offers any adult-education courses in computers. Some towns offer adult-education computer courses for under $100.

For an even better deal, phone your town's board of education— or high school— and ask whether you can sit in the back of a high-school computer class. If you're an adult resident of the town, you might be able to sit in the back of the class for free. Your only "expense" will be the embarrassment of sitting in the same room as youngsters. After a day or two of feeling strange, you'll get used to it, and you'll get an excellent education, free.

Community colleges offer low-cost courses that are decent. Explore the community colleges before sinking money into more expensive institutions that are over-priced.

The best course of all is the one I teach during the summer. It starts at the very beginning, is super-intensive, and costs you practically nothing. (It's run by a bunch of computer missionaries, who don't care much about money.) Because we use powerful psychological techniques, the students who come out of the course often get jobs as systems analysts, editors of computer magazines, heads of computer centers, vice-presidents of computer companies, etc.— even though the students knew little-or-nothing about computer before the course began, and even though the course lasts only seven weeks! The course includes one-to-one tutoring, 24 hours, throughout the seven weeks, and can cost as little as $125. We can even give you housing, for almost free. For full details about the course, read the last chapter of this volume.

Starting salary

For your first programming job, your salary will be somewhere between $10,000 and $20,000. The exact amount depends on which languages you know, how many programs you wrote previously, whether you have a college degree, whether you've had experience on the particular kind of computer the company uses, and whether you know the application area. (For example, if you're a programmer for an insurance company, it's helpful to know something about insurance.)

Degrees

A college degree is not essential, but helpful. Try to get a degree in "computer science" or "management information systems". "Computer science" emphasizes the underlying theory, systems programming, assembly language, PASCAL, FORTRAN, and applications to science; "management information systems" emphasizes BASIC, COBOL, and applications to business. A major in "mathematics" that emphasizes computers is also acceptable.

Discrimination

If you're a woman or non-white or physically handicapped, you'll be pleased to know that the computer industry discriminates less than in other occupations. In fact, being a woman or non-white or physically handicapped works to your *advantage*, since many companies have affirmative-action programs.

On the other hand, discrimination *does* exist against older people. If you're over 40 and trying to get a job as an entry-level programmer, you'll have a tough time, since the stereotypic programmer is "young, bright, and a fast thinker". If you're old, they'll assume you're "slow and sluggish".

Because of that unfair discrimination, if you're old you should probably try entering the computer industry through a different door: as a consultant, or a computer salesperson, or a computer-center manager, or a computer teacher. For those positions, your older age works to your *advantage*, since those jobs require *wisdom*, and people will assume that since you're old, you're wise.

Shifting careers

If you're older, the best way to enter the computer field is to combine your new knowledge of computers with what you knew previously. If you already knew a lot about sales, get a job selling computers. If you already knew a lot about teaching, get a job teaching about computers— or helping teachers deal with computers. If you already knew a lot about real estate, computerize your real estate office.

In other words, do *not* try to "hop" careers; instead, gradually *shift* your responsibilities so that they deal more with computers.

To get into the computer field safely, retain your current job but computerize it. For example, if you're already a math teacher, keep teaching math but also convince your school to let you teach a computer course also, or at least incorporate computers into the math curriculum, or at least help run the school's computer center. If you already work for a big company and are doing a boring job, try to get transferred to a different department and which puts you in closer contact with the computer. After a year in such a transitional state, you can break into the computer field more easily, since you can put the word "computer" somewhere on your resumé, as "job experience".

If you're in college kid, write programs that help the professors, or help others during your summer vacations. Agree to write the programs for little or no pay. Your goal is *not* money: your goal is to put "experienced programmer" on your resumé.

Interviews

When applying for your first computer job, avoid the "personnel" office. The people in the personnel office will look at your resumé, see it includes too little experience, and trash it.

Instead, play the who-you-know game. Contact somebody who actually works with computers, convince that person you're brighter than your resumé indicates, and prove to that person you've learned so much (from reading, courses, and practice) that you can conquer any task laid before you. Convince that person you can think quickly. If you impress that person enough, that person can get you a job, even though your paper qualifications look too brief.

When you get an interview, be aggressive: assert yourself. Ask the interviewer more questions than the interviewer asks you. Ask the interviewer about the company's computer, and about why the company doesn't have a different one instead. Ask the interviewer how the other people in the company feel about the computer center. Ask the same kinds of questions a data-processing manager would ask. That way, the interviewer will assume you have the potential to become a data-processing manager, and will hire you immediately. You'll also be showing you *care* enough about the company to ask questions. And you'll be showing you have a vibrant personality, and are not just "another vegetable who came through the door".

One of the strange things about applying for a programming job is that the interviewer will *not* ask to see a sample of your work. The interviewer doesn't have time to read your program. Even if the interviewer *did* have time to read your program, he couldn't be sure you wrote it yourself. Instead, the interviewer will just *chat* with you about your accomplishments. So you must "talk smart". The best way to "talk smart" is to know all the buzzwords of the computer industry— even if they don't really help you write programs. Volume 3 explained fundamental buzzwords: reread it.

During the interview, you'll probably be asked whether you know "structured programming". A *structured program* is a program that's well-organized. It consists of a short main routine and many subroutines. To write a structured program, avoid the words GO TO; instead, use words that involve subroutines:

Language	Words that involve subroutines
BASIC	GO SUB and RETURN
FORTRAN	CALL, SUBROUTINE, and RETURN
PASCAL	PROCEDURE
COBOL	PERFORM

Later joys

In your first job, your salary will be low, but don't worry about it. During your first job, you'll receive lots of training: you're getting a free education. After your training period is over, your salary will rise rapidly— especially if you do extra studying during evenings and weekends. Your *real* job is: to become brilliant.

After you've become brilliant and experienced, other companies will eagerly want to hire you. Your best strategy is to leave your current company and work elsewhere, to gain new experiences. Whenever you feel you're "coasting", and not learning anything new, it's time to move to a different job. The "different job" can be in a new company— or in a different department of the current company.

By moving around— by gaining a wide variety of experiences— you can eventually become a qualified, wise consultant. And you'll feel like "King of the World".

Social contacts

Being a programmer is not always glamorous. You'll spend many long hours staring at your terminal's screen and wondering why your program doesn't work. The job is intellectual, not social. But after you've become an expert coder, you get into "systems analysis" and "consulting" and "teaching" and "management", and interact with more people.

Software publishing

To be a programmer, you do *not* have to work for a large company. Instead, you can sit at home, write programs on your personal microcomputer, and sell them to software publishers, for a royalty. If the software publisher sells many copies of your program, you become rich. (On the other hand, if your program is *not* a "smash hit", you remain poor.)

Since your program might not become popular, do *not* rely on software publishing as a steady source of income. Instead, view it as a part-time activity which, if successful, will put some extra money in your pocket.

The most famous software publishers are Instant Software, Hayden, and Creative Computing; and there are many others. Browse through the ads in microcomputer magazines.

Software houses

A company whose only goal is to produce software is called a *software house*. You can work for a software house full-time (for a fixed salary) or for a royalty. If the software house deals with large computers, the usual method of payment is full-time fixed-salary, rather than royalty.

MANAGEMENT

Programming is fun for young kids. But as you get older, you'll tire of machines, and want to deal with people instead. As you approach retirement, you'll want to help the younger generation relate to the computers you've mastered.

To be a successful manager, you need three skills: you must be technically competent; you must be wise; and you must know how to handle people.

You should know how to program. You should know the strengths and weaknesses of each computer company, and be able to compare their products. You should have a philosophy about what makes a "good" computer center. You should understand people's motives, and channel them into constructive avenues. You should keep up-to-date, by reading the latest books and periodicals about computers, and by chatting with other computer experts via phone and at conventions. If you live in New England and care about microcomputers, join the Boston Computer Society (3 Center Plaza, Boston, MA 02108, 617-367-8080).

When trying to run a computer center, you can easily make mistakes. For example, many computer centers put four-foot-high partitions between their programmers, to give the programmers "privacy"; unfortunately, the partitions are counter-productive: they're too low to block noise, and too high to permit helpful conversation with your neighbor.

If you're putting a computer center into a high school (or junior high or middle school or even an elementary school), you must develop a *cadre* of hot-shot students who are bright, friendly, and outgoing, and who will help and encourage the other students to use the computer. If the hot-shots are *not* outgoing— if they become an elitist, snobbish club— the rest of the school will avoid the computer.

If you've hired "programming assistants" who help the programmers, don't let the "programming assistants" hide in an office or behind a desk; tell the programming assistants to go to the terminals, chat with the programmers, and ask the programmers whether the programmers could use any help.

In too many organizations, terminals are locked in the offices of prestigious people and aren't used. Let *everybody* share the terminals.

Too often, managers judge their own worth by the size of the computer center's budget: the bigger the budget, the more prestigious the manager. Remember that the sign of being a good manager is *not* having a big budget; the sign of a good manager is the ability to meet the company's needs on a *small* budget.

In computer jargon, a *word* is how many bits the CPU can handle simultaneously. For most microcomputers, a word is 8 bits (i.e., the CPU can handle 8 bits simultaneously); for most minicomputers, a word is 16 bits; for most maxicomputers, a word is 32 bits. A CPU having a big word is prestigious, but might not be cost-effective.

Too often, the head of the computer center decides who can use the computer. So the head of the computer center becomes powerful— and evil. To avoid concentrating so much power in the hands of one bureaucrat, use *distributed processing*: get several small computers instead of one big monster, and give each department its own small computer.

If you're a "microcomputer consultant" and honest, you'll tell your client to buy low-cost popular programs, instead of telling him to pay you to invent "customized" programs.

SALES

You can find three kinds of salesmen.

The "slick" kind knows "how to sell", but doesn't know any technical details about the computer he's selling; he doesn't know how to program, and doesn't know much about the computers sold by his competitors. All he knows is the "line" that his boss told him to give the customers. That kind of salesman usually resorts to off-color tactics, such as claiming that all computers sold by competitors are "toys".

The opposite kind of salesman is technical: he knows every detail about every computer manufactured, but can't give you any *practical* advice about which computer best meets *your* needs.

The best kind of salesman is a consultant: he asks lots of questions about your particular needs, tells you which of his computers meets your needs best, and even tells you the *limitations* of his computer and why another, more expensive computer sold by a competitor might be better. He's an "honest Joe". He clinches the sale because you trust him, and because you know you won't have any unpleasant surprises after the sale. While selling you a computer, he teaches you a lot. He's a true friend.

A woman can sell computers more easily than a man. That's because most computer customers are men, and men are more attracted to women. It's also because, in our society, women are more "trusted" than men. But if you're a woman, say some technical buzzwords, to convince the customer that you're technically competent; otherwise, the customer will assume that since you're a woman you must be a "dumb secretary".

ORIGINAL JOBS

How about starting a rental service, where people can rent microcomputers? How about starting a camp, where kids can spend the summer playing with computers? How about starting a computer set-up service, where you and your students help executives get started using their Apple and Radio Shack computers? How about writing easy-to-use explanations about how to select and use the most popular computers? Each of those ideas has been tried successfully; join the fun!

History

BEFORE COMPUTERS

The first programmable computers were invented in the 1940's. Before then, there wasn't much to talk about. People had invented the abacus, then the adding machine, and then the slide rule, but those gadgets were far simpler than computers.

The history of punched cards is weird. The cards were first used for *weaving tapestries*. Where the cards had holes, rods could move through the cards; those moving rods in turn made other rods move, which caused the threads to weave pictures. That machine was called the *Jacquard loom*.

CHARLES BABBAGE

Charles Babbage was a wild-eyed English mathematician who, in the 1800's, believed he could build a fancy computing machine. He convinced the British government to give him lots of money. Then he bilked the government for more. Many years later— and many British-pounds later— he still hadn't finished his machine. So he dropped the idea and— can you believe this?— tried to build an even *fancier* machine. Needless to say, he didn't finish that one either. So you might say his life was a failure— and an expensive one, for the British government.

But Charlie (as I shall call him) is admired by all us computerniks (in spite of his face, which was even sterner than Beethoven's), because he was the first person to realize that a computing machine must be composed of an input device (he used a card reader), a memory (which he called "The Store"), a central processing unit (which he called "The Mill"), and an output device (he used a printer).

LADY LOVELACE

Women's Libbers will kill me if I don't mention Charlie's side-kick, Lady Lovelace. (No, she's not related to Linda.) She was one of Charlie's great admirers, but he never noticed her until she translated his stuff. And boy, it was impossible for him *not* to notice her translation. Her "footnotes" to the translation were three times as long as what she was translating!

She got very intense. She wrote to Charlie, "I am working very hard for you; like the Devil in fact; (which perhaps I am)."

The two became love-birds, although he was old enough to be her father. (By the way, her father was Lord Byron, the poet. She was Lord Byron's only "official" daughter. He had several other daughters, but they were illegitimate.) Some people argue that she was actually brighter than Charlie, despite Charlie's fame. She was better at explaining Charlie's machines and their implications than Charlie was. Some people have dubbed her, "the world's first programmer".

She stunned all the men she met. She was so bright and... a woman! Here's how the editor of *The Examiner* described her (note the pre-Women's-Lib language!):

"She was thoroughly original, and her genius, for genius she possessed, was not poetic, but metaphysical and mathematical. With an understanding thoroughly masculine in solidity, grasp, and firmness, Lady Lovelace had all the delicacies of the most refined female character. Her manners, her tastes, her accomplishments, were feminine in the nicest sense of the word; and the superficial observer would never have divined the strength and the knowledge that lay hidden under the womanly graces. Proportionate to her distaste for the frivolous and commonplace was her enjoyment of true intellectual society. Eagerly she sought the acquaintance of all who were distinguished in science, art, and literature."

Eventually, she went mad. To prevent her from banging her head, they lined her room with mattresses. Nevertheless, she died gruesomely, at the ripe young age of 36, the same age her father died. (I guess premature death was popular in her Devilish family.)

I wish the Women's Lib press would pick a different idol than Lady Lovelace. She was *not* the most important woman in the history of computing. Far more important were Grace Hopper (who, in the 1950's, invented the first programming languages) and Jean Sammet (who headed the main committee that invented COBOL, and who is generally considered to be *the* world's expert on the history of programming languages, and who has been president of the U.S. organization of computer professionals, which is called the ACM). Lady Lovelace was second-string to Babbage. Grace Hopper and Jean Sammet are second-string to *nobody*. But unfortunately, Hopper and Sammet lead less racy lives, so journalists ignore them. Also, since Hopper is an Admiral in the Navy (bet you didn't know the Navy has lady Admirals!), she doesn't enchant some of us doves. Nevertheless, every time she steps in front of an audience, she gets a standing ovation, because all of us realize how crucial she was to what computers are today.

But I'm straying from my story....

HERMAN HOLLERITH

The U.S. Bureau of the Census takes its census every ten years. For the 1880 census, the guys at the Bureau had a hard time: it took them *7 years* to tabulate the results of the census. In other words, they didn't finish until 1887. When they contemplated the upcoming 1890 census, they got scared: at the rate America was growing, they figured that tallying the 1890 census would take 12 years. In other words, the results of the 1890 census wouldn't be ready until 1902. So they held a contest, to see whether anyone could invent a faster way to tabulate the data.

The winner was Herman Hollerith. He was the first person to successfully use punched cards to process data.

Hermie (as I shall call him) was modest. When people asked him how he got the idea of using punched cards, he had two answers. One was, "Trains": he had watched a train's conductor punch the tickets. His other, more interesting answer was, "Chicken salad". After saying "Chicken salad", he would pause, and wait for you to ask the obvious question, "*Why* chicken salad?" Then he would tell his tale....

One day, a girl saw him gulping down chicken salad. She said, "Oh, you like chicken salad? Come to my house. My mother makes excellent chicken salad." So he did. And her father was a head of the Census. (And he married the girl.)

By the way, Herman Hollerith hated one thing: spelling. In elementary school, he jumped out a second-story window, to avoid a spelling test.

In some versions of FORTRAN, every string must be preceded by the letter H. For example, instead of saying—

```
'DOG'
```

you must say:

```
3HDOG
```

The H is to honor Herman Hollerith.

Anyway, getting back to the story, in 1890 the Census used Hollerith's system, and used it again in 1900. But in 1910, the Census switched to a more advanced system, created by a Census Bureau employee named James Powers, who eventually left the Census to start his own company, which eventually merged into Remington-Rand-Sperry-Univac. Meanwhile, Herman Hollerith's own company merged into IBM. And that's how the first two computer companies got started in data processing.

WORLD WAR II

The first programmable computers were invented in the 1940's, because of World War II. They *could* have been invented sooner— most of the know-how was available several decades earlier— but you can't invent a computer unless you have big bucks for research. And the only organization that had big enough bucks was the Defense Department (which in those days was more honestly called the "War Department"). And the only event that was big enough to make the War Department spend that kind of money was World War II.

Of course, the Germans did the same thing. A German fellow named Konrad Zuse built computers which were in some ways better than the American ones. But since the Germans lost the war, you don't hear much about old Konrad anymore. Fortunately, throughout World War II the German military ignored what he was doing.

Most of the computers in the 1940's were invented at universities, usually with funds from the War-Defense Department. Some of the most famous computers were the Mark I (at Harvard, with help from IBM), the ENIAC and the EDVAC (both at the University of Pennsylvania), the Whirlwind (at M.I.T.— the Massachusetts Institute of Technology), and the Ferranti Mark I (at the University of Manchester, in England). Which of those computers deserves to be called "the first programmable computer"? The answer's up for grabs. Each of those machines had its own peculiar hang-ups, and required several years of debugging before it actually worked decently.

Each of those computers was, as they say in the art world, a "signed original". No two of those computers were alike.

THE FIRST GENERATION
1951-1958

It all started with UNIVAC

The first computer to be mass-produced with the UNIVAC I, in 1951. It was made by two guys, named Eckert and Mauchly— the same guys who built the ENIAC and EDVAC at the University of Pennsylvania. (Mauchly was an instructor there, and Eckert was the graduate student who did the dirty work.) While others at the school were helping build the EDVAC, Eckert and Mauchly left and formed their own company, which invented and started building the UNIVAC. But by the time the UNIVAC was completed, the Eckert-Mauchly company had merged into Remington Rand (which later merged into Sperry-Rand, which sprouted a division which today is called Sperry-Univac).

Because the UNIVAC I was so important, historians call it the beginning of the "first generation". As for computers before UNIVAC— historians disparagingly call them the "zeroth generation".

So the first generation began in 1951. It lasted through 1958. Altogether, from 1951 to 1958, 46 of those UNIVACs were sold.

46 might not sound like many. But remember: in those days, computers were very expensive, and could do very little. Another reason why only 46 were sold is that newer models came out, such as the UNIVAC 1103, the UNIVAC 80, and the UNIVAC 90. But the biggest reason why only 46 of the UNIVAC I were sold is IBM.

The rise of IBM

Although IBM didn't begin mass-marketing computers until 1953— two years after UNIVAC— the IBM guys were much better salesmen, and soon practically everybody was buying from IBM. During the first generation, the hottest seller was the IBM 650. Yep, IBM sold hundreds and hundreds of them.

Of course, there were many smaller manufacturers too. People summarized the whole computer industry in one phrase: "IBM and the Seven Dwarfs".

Who were the dwarfs? They kept changing. Companies rapidly entered the field and then rapidly left, when they realized IBM had the upper hand. By the end of the first generation, IBM was getting 70% of the sales.

Primitive input and output

During the first generation, there were no terminals. To program the UNIVAC I, you had to put the program onto magnetic tape (by using a non-computerized machine), then feed the tape into the computer, then wait for the computer to spit out another magnetic tape, which you then had to run through another machine, to find out what the tape said.

One reason why the IBM 650 became more popular was that it could read cards instead of tapes. It really liked cards. In fact, the answers came out on cards. To transfer the answers from cards to paper, you had to run the cards through a separate non-computerized machine.

Memory

At the beginning of the first generation, there was no RAM, no ROM, and no core. Instead, the UNIVAC's main memory was banks of liquid mercury, in which the bits were stored as ultrasonic sound waves. It worked slowly and serially, so the access time ranged from 40 to 400 microseconds per bit.

UNIVAC's manufacturer and IBM started playing around with a different kind of memory, called the Williams tube, which was faster (10 to 50 microseconds); but since it was less reliable, it didn't sell well.

In 1953, several manufacturers started selling computers that were much cheaper, because they used super-slow memory: it was a drum that rotated at 3600 rpm, giving an average access time of 17000 microseconds (17 milliseconds). (Today, some computers still use drums, but for *auxiliary* memory, not for *main* memory.) The most popular first-generation computer, the IBM 650, was one of those cheap drum computers.

The idea of core memory began in 1950. The first working models were invented in 1953 at M.I.T. and RCA, which argued with each other about who owned the patent. The courts decided in favor of M.I.T., so both RCA *and IBM* came out with core-memory computers. Core memory proved so popular that even today many minicomputers and maxicomputers still use core, though RAM is finally starting to overshadow it.

Languages

During the first generation, computer programming improved a lot. During the early '50s, all programs had to be written in machine language. In the middle 50's, assembly language became available. By 1958, the end of the first generation, three major high-level languages had become available: FORTRAN, ALGOL, and APT.

Fancy programs

Programmers tried to make computers play a decent game of chess. All the attempts failed. But at IBM, Arthur Samuel had some luck with checkers. He got his first checkers program working in 1952, and then continually improved it, to make it more and more sophisticated. In 1955, he rewrote it so that it learned from its own mistakes. In 1956, he demonstrated it on national television. He kept working on it. Though it hadn't reached championship level yet, it was starting to look impressive.

Computer music scored its first big success in 1956, on the University of Illinois' ILLIAC computer. Hiller & Isaacson made the ILLIAC compose its own music, in a style that sounded pre-Bach. In 1957, they made the program more flexible, so that it produced many styles of more modern music. The resulting mish-mash composition was dubbed "The ILLIAC Suite", and put onto a phonograph record.

In 1954, IBM wrote a program that translated simple sentences from Russian to English. Work on tackling harder sentences continued— with too much optimism.

THE SECOND GENERATION
1959–1963

Hardware

Throughout the first generation, each CPU was composed of vacuum tubes. Back in 1948, Bell Telephone had invented the transistor, and everybody realized that transistors would be better than vacuum tubes; but putting transistors into computers posed many practical problems, which weren't solved for many years. Finally, in 1959, computer companies suddenly started delivering transistorized computers; that year marked the beginning of the second generation; sales of vacuum-tube computers immediately stopped.

All second-generation computers used core memory.

IBM

The *first* company to make transistors for computers was Philco, but the most *popular* second-generation computer turned out to be the IBM 1401, because it was business-oriented and cheap. IBM announced it in 1959, and started shipping it to customers in 1960.

Its core memory required 11.5 microseconds per character. Each character consisted of 6 bits. The number of characters in the memory could range from 1.4K up to 16K. Most people rented the 1401 for about $8,000 per month, but you could spend anywhere from $4,000 to $12,000 per month, depending on how much memory you wanted, etc.

Altogether, IBM installed 14,000 of those machines. IBM also installed 1,000 of a faster version called the 1410 (which required only 4.5 microseconds per character, had 10K to 80K, and rented for $8,000 to $18,000 per month, typically $11,000).

Even today, many of those 1401 and 1410 machines are hanging around, and companies are still trying to convert programs from the 1401 to the IBM 370.

Altogether, IBM produced six kinds of computers....

small business computers: the 1401, 1410, 1440, and 1460
small scientific computers: the 1620

medium-sized business computers: the 7010
medium-sized scientific computers: the 7040 and 7044

large business computers: the 7070, 7074, and 7080
large scientific computers: the 7090 and 7094

CDC

Several employees left Remington-Rand-Sperry-Univac and formed their own company, called the Control Data Corporation (CDC). During the second generation, CDC produced popular scientific computers: the 1604, the 3600, and the 3800.

Software

During the second generation, software improved tremendously.

The 3 major programming languages that had been invented during the first generation (FORTRAN, ALGOL, and APT) were significantly improved.

6 new programming languages were invented: COBOL, RPG, LISP, SNOBOL, DYNAMO, and GPSS.

The 3 most popular sorting methods were invented: the Shuffle Sort, the Shell Sort, and Quicksort.

Programmers wrote advanced programs that answer questions about baseball, write poetry, tutor medical students, imitate three-person social interaction, control a mechanical hand, prove theorems in geometry, and solve indefinite integrals.

THE DAWN OF THE THIRD GENERATION
1964-1967

The big bang

The third generation began with a big bang, in 1964. Here's what happened in 1964, 1965, 1966, and 1967....

Families

The first modern computer families were shipped. They were the CDC 6600, the IBM 360, the PDP-6, the PDP-8, and the PDP-10.

Of those families, the CDC 6600 ran the most quickly. The IBM 360 was the most flexible, and was also the only one that used integrated circuits. The PDP-6 and PDP-10 were the best for timesharing. The PDP-8 was the cheapest.

Here are the exact dates. In 1964, CDC began shipping the CDC 6600. In 1964, IBM announced it was going to create a new family, the IBM 360; shipments began in 1966. In 1964, DEC began shipping the PDP-6 (a maxicomputer); in 1965, DEC began shipping the PDP-8 (a minicomputer); in 1967, DEC began shipping the PDP-10 (a maxicomputer that's a souped-up PDP-6).

New languages

IBM announced it was going to create a new computer language, that would combine all previous popular languages, including FORTRAN, COBOL, and ALGOL. That new language was PL/I. It was designed especially for IBM's new computer, the 360. In 1966, IBM began delivering PL/I to customers.

Programmers invented the first successful languages for *beginners* using *terminals*. Those languages were called BASIC, JOSS, and APL. Here are the exact dates....

Dartmouth College invented the first version of BASIC in 1964, and significantly improved it in 1966 and 1967.

In 1964, the RAND Corporation invented JOSS, for the JOHNNIAC computer. In 1965, an improved version, called JOSS II, was put onto the PDP-6. Today people use three popular variations of JOSS: a souped-up version (called AID), a stripped-down version (called FOCAL), and a business-oriented version (called MUMPS).

IBM completed the first version of APL in 1965, and put it on an IBM 7090. In 1966, IBM wrote a much better version of APL, and put it on an IBM 360. In 1967, IBM began shipping APL to customers.

Stanford University invented the most popular language for statistics: SPSS.

Artificial intelligence

Researchers (who called themselves "experts in artificial intelligence") taught the computer to chat in ordinary English:

Bertram Raphael made the computer learn from conversations.
Daniel Bobrow made it use algebra to solve "story problems".
The Systems Development Corporation made it know everything in an encyclopedia.
General Electric made it answer military questions.
Ross Quillian made it find underlying concepts.
Joe Weizenbaum made it act as a psychotherapist.

Also, Richard Greenblatt wrote the first decent chess program. It was good enough to play in championship tournaments against humans.

THE ERA OF BOREDOM
1968-1974

Proof of boredom

As you can see, the first, second, and third generations— up through 1967-- were exciting, full of action. But then, from 1968 to 1974, *nothing newsworthy happened*. That was the era of boredom.

During the era of boredom, progress was made, but it was gradual and predictable. Nothing dramatic happened.

Of course, nobody actually came out and said, "Life is boring." People phrased it more genteelly. For example, this passage was written in September 1971, by Robert Fenichel and Joe Weizenbaum, for their introduction to *Scientific American*'s computer anthology:

"Partly because of the recent recession in the American economy, but more for reasons internal to the field, computer science has recently relaxed its pace. Work has not stopped, but that the current mood is one of consolidation can scarcely be doubted. Just a few years ago, computer science was moving so swiftly that even the professional journals were more archival than informative. This book could not then have been produced without great risk of misfocus. Today it is much easier to put the articles that constitute this book— even the most recent ones— into context."

Since the first generation had lasted eight years (1951-1958), and the second generation had lasted four years (1959-1963), so people were expecting the third generation to last at most four years (1964-1967) and that some kind of "fourth generation" would begin about 1968. But it never happened.

The only "major" announcement around then came in 1970, when IBM announced it would produce a new line of computers, called the IBM 370, which would make the IBM 360 obsolete. But to IBM's dismay, many computer centers decided to hang onto the old 360, instead of switching to the 370. The 370's advantage over the 360 was little, until several years later, when IBM started developing 370 software that wouldn't run on the 360.

Since the difference between the 370 and the 360 was disappointingly small, not even IBM claimed that the 370 marked a fourth generation. Computer historians, desperate for something positive to say about the 370, called it the beginning of the "late third generation", as opposed to the 360, which belonged to the "early third generation".

The cruel fact is, in the entire history of computers, there was only one year all computer manufacturers acted together to produce something new. That year was 1959, when all manufacturers switched from vacuum tubes to transistors. Since 1959, we haven't had any consistency. For example, although the third generation began with a "big bang" in 1964, each manufacturer was banging on a different drum. IBM was proclaiming how great the IBM 360 would be, because it would contain integrated circuits; but other manufacturers decided to ignore integrated circuits for several eayrs, and concentrated on improving other aspects of the computer instead. For many years after the beginning of the third generation, CDC and DEC continued to use discrete transistors (a sign of the second generation) instead of integrated circuits.

Why?

The era of boredom happened for three reasons....

The preceding years, 1964-1967, had been so successful that they were hard to improve on.

When the Vietnam War ended, the American economy had a recession, especially the computer industry, because it had depended on contracts from the Department of Defense. In 1969, the recession hit its worst point, and computer companies had to lay off many workers. In that year, General Electric gave up, and sold its computer division to Honeywell. In 1971, RCA gave up also, and sold its computer division to Remington-Rand-Sperry-Univac.

The world wasn't yet ready for "the era of personal computing", which began in 1975.

Quiet changes

During the era of boredom, these changes occurred— quietly....

In 1970, DEC began shipping the PDP-11. The PDP-8 and PDP-11 became the most popular minicomputers— far more popular than IBM's minicomputers. So in the field of minicomputers, IBM no longer had the upper hand.

BASIC became the most popular language for the PDP-8 and the PDP-11 and most other minicomputers (except IBM's, which emphasized RPG). In high schools and business schools, most of the introductory computer courses used BASIC, instead of FORTRAN or COBOL.

Many businesses and high schools bought their own minicomputers, instead of renting time on neighbors' maxicomputers. The typical high-school computer class used a PDP-8. The richest high schools bought PDP-11's.

In universities, the social sciences started using computers— and heavily— for doing statistical analysis.

All new computer families used 8-bit bytes, so the length of each word was a multiple of 8 (such as 8, 16, 32, or 64). Most older computer families, invented before the era of boredom, had used 6-bit bytes, so the length of each word had been a multiple of 6: for example, the PDP-8 had a word of 12 bits; the PDP-10 and UNIVAC 1100 and General Electric-Honeywell computers had a word of 36 bits; and the CDC 6600 had a word of 60 bits. The IBM 360 was the first computer to use 8-bit bytes instead of 6-bit; during the era of boredom, all manufacturers copied that feature from IBM.

CRT terminals (which use T.V.-like screens) got cheaper and cheaper, until they were finally as cheap as hard-copy terminals (which use paper). Most computer centers replaced hard-copy terminals by CRT terminals, since CRT terminals were quicker, quieter, and could do fancy editing.

Use of keypunch machines decreased, because many computer centers replaced cards by CRT terminals.

Interest in new computer languages died. Most computer managers decided to stick with the old favorites— FORTRAN and COBOL— instead of switching to more progressive languages, such as PL/I. Switching to PL/I would have required many programmers to be retrained, and many programs to be rewritten; managers couldn't afford that.

Programmers made two last-ditch attempts to improve ALGOL. The first attempt, called "ALGOL 68", failed to win popular appeal, because it was too complicated. The second attempt, called "PASCAL", gained more support, but slowly.

Maxicomputers were given "virtual core"— disks that pretend to be core, in case you're trying to run a program that's too large to fit into core.

Semiconductor memory got cheaper and cheaper, until it was finally cheaper than core. Most manufacturers replaced core by semiconductor memory.

In 1971, Intel began shipping the first microprocessor (complete CPU on a chip). It was called the 4004, and had a word of only 4 bits. In 1972, Intel began shipping an improved version, called the 8008, whose word had 8 bits. In 1973, Intel began shipping an even better version, called the 8080.

THE ERA OF PERSONAL COMPUTING
1975 and afterwards

Altair

In January 1975, the cover of *Popular Electronics Magazine* featured a "complete computer for only $395". That computer was manufactured by a company called "MITS", which called the computer the "Altair 8800", because Altair is the main star in Star Trek. Many people read the article and bought the computer.

By today's standards, the computer was *not* complete. It came as a kit you had to assemble yourself, by using a soldering iron and following the wiring diagrams. It included *no* software— not FORTRAN, not BASIC, not even assembly language or an editor or an operating system. The only way to program it was to use machine language, and toggle in the program by flipping the switches on the front panel. The only way to get the answers was to look at the blinking lights, which gave the answers in binary. There was no ROM, and only ¼ of a K of RAM. There was no disk, no tape, no printer, and no terminal— not even a keyboard. When you bought the kit and tried to put it together, you usually found that parts were missing, and the diagrams contained errors. In short, it was just an annoying toy, for tinkerers. Moreover, MITS quickly raised the price from $395 to $439.

Gradually, MITS fixed its errors, and began offering software, RAM, and peripherals— at extra cost, so that the typical customer wound up paying several thousand dollars extra to get the computer to do anything useful. But what's a few thousand dollars, when most competitors were charging *tens* of thousands of dollars?

MITS got lots of attention from the press, which thought it might lead to "a computer in every home".

Imitations

Several new companies arose a tried to steal MITS's business, by imitating MITS. They offered equipment that was similar to MITS's but better and lower-priced. The most famous of these competitors was IMSAI, which bragged that its CPU, memory, etc., were interchangeable with MITS's but were more "rugged" and "durable" and cost less. Other early competitors were Processor Technology and Polymorphic Systems.

Just like MITS, all those companies used the 8080 CPU and the S-100 bus and offered BASIC, though the versions of BASIC were not quite the same. The hobbyist could do comparative shopping, buy different parts of his computer from different companies, and get the whole combination to work.

A few manufacturers tried to be different: Southwest Technical Products used the 6800 CPU; MOS Technology's KIM used the 6502 CPU; the Digital Group used the 8080 CPU but with an unusual bus. Those manufacturers fared less well than the ones that imitated the Altair.

The biggies

In 1977, two larger companies got into the act: Commodore announced it was going to make a computer called the "PET"; and Radio Shack announced it was going to make a computer called the "TRS-80". Today, the most popular personal computers are Radio Shack's TRS-80, the Apple 2-plus, and Commodore's computer (which comes in several models: the VIC, the PET, and the CBM).

Spelling

ERRORS

Beginners often misspell these words:

Wrong	Right	Comment
COBAL	COBOL	"COBOL" means "COmmon Business-Oriented Language". "COBAL" is a co-ed who likes sex.
computor	computer	"Computer" is a machine or person that computes. "Computor" is a snobbish computer.
Dartmouth U.	Dartmouth Coll.	"Dartmouth Coll." is where BASIC was invented. "Dartmouth U." exists only in Utopia.
Epsom	Epson	"Epson" makes printers. "Epsom" makes salt.
hexidecimal	hexadecimal	"Hexadecimal" means "six and ten", which is "sixteen". "Hexidecimal" is icky.
hobbiest	hobbyist	A "computer hobbyist" is a person who likes computers. A "computer hobbiest" is even more hobbier.
imput	input	"Input" is what the computer takes "in". "Imput" is said only by "im"beciles.
silicone	silicon	"Silicon" is what you put in an integrated circuit. "Silicone" is what you put in your breast.
softwear	software	"Software" is the opposite of "hardware". "Softwear" is a negligée.
TRS-232	RS-232	"RS-232" means "Recommended Standard #232". "TRS-232" means you think Tandy's Radio Shack is God.
TSR-80	TRS-80	"TRS-80" stands for "Tandy's Radio Shack". "TSR-80" is a jerk who mumbles the alphabet backwards.

CHOICES

For these words, choose your favorite spelling:

IBM prefers "disc"; other companies prefer "disk".
To mean "half a byte", humble programmers prefer "nibble"; snobs prefer "nybble".
The British prefer "programer"; Americans prefer "programmer".

Books

HOW TO USE THE BOOK LIST

After you've read all 8 volumes of The Secret Guide to Computers, you'll want to read other books also, by other publishers. Here's a list of the best books.

For each topic, I've listed two books; if you can afford to buy both books, read book #1 before book #2. For each book, I've listed the title, author, price, and publisher. Because prices go up each year, I've rounded each price to the nearest $5.

For a well-rounded easy education at low cost, buy each book that has 3 stars. If you can afford to spend more, also buy each book that has 2 stars. Then the books that have 1 star...then the others.

BOOKS THAT ELABORATE ON VOLUMES 1 & 2 (BASIC)

How to program, by using BASIC

*** #1 *BASIC and the Personal Computer* by Dwyer & Critchfield ($15 Addison-Wesley)
*** #2 *Programming in Style* by Dwyer & Critchfield ($10 Radio Shack)
#1 is fun; #2 is sophisticated. #1 emphasizes the PDP-11; #2 emphasizes the TRS-80.

Programs, written in BASIC, that play games and that you can copy

#1 *BASIC Computer Games (TRS-80 edition)* by Ahl ($5 Radio Shack)
#2 *BASIC Computer Games Volume 2 (TRS-80 edition)* by Ahl ($5 Radio Shack)
#1 contains 102 games. #2 contains 84.

Details about Radio Shack's TRS-80 Level-2 BASIC

* #1 *Programming Techniques for Level 2 BASIC* by Barden ($5 Radio Shack)
* #2 *Learning Level 2* by Lien ($15 Compusoft)
#1 is inexpensive. #2 is easier, covers more commands, but is superficial.

Instructional aids

*** #1 *The Little Book of BASIC Style* by Nevison ($5 Addison-Wesley)
#2 *A Collect'n of Prog'ming Problems & Techniq's* by Maurer & Williams ($15 Pr.-Hall)
#1 improves your programming style. #2 gives you exercises to try.

BOOKS THAT ELABORATE ON VOLUMES 3 & 4 (SYSTEMS)

Microcomputers

* #1 *Owning Your Home Computer* by Perry ($10 Everest)
** #2 *Business System Buyer's Guide* by Osborne & Cook ($10 Osborne)
#1 emphasizes fun applications; #2 emphasizes business. (Each contains a few errors.)

Larger computers

*** #1 *Introduction to Computers & Data Processing* by Shelly & Cashman ($15 Anaheim)
** #2 *Encyclopedia of Computer Science* by Ralston & Meek ($60 Petrocelli/Charter)
#1 tutors you. #2 is a huge encyclopedia & costs just $3 from book clubs.

BOOKS THAT ELABORATE ON VOLUMES 5 & 6 (APPLICATIONS)

Survey of applications

** #1 *Computers in Society* by Spencer ($5 Hayden)
** #2 *Computer Lib/Dream Machines* by Nelson ($10 Hugo's Book Service)
#1 shows the good things computers are doing. #2 is cynical, nasty, & funny.

Directories, that tell you where to buy software

#1 *Applications Software Sourcebook* by Radio Shack ($5 Radio Shack)
#2 *The Book of Apple Computer Software* by Sadlier & Stanton ($20 The Book Co.)
#1 is for Radio Shack's computers. #2 is for Apple, and includes criticisms of software.

Creating emotional art

#1 *Cybernetic Serendipity* by Reichardt (out of print, borrow it, Praeger)
#2 *Artist & Computer* by Leavitt ($5 Creative Computing)
#1 covers all arts; #2 covers just graphics. #1 is older (1969) but more interesting.

Techniques & equipment for 3-dimensional perspective drawing

* #1 *Principles of Interactive Computer Graphics* by Newman & Sproull ($25 McGraw-Hill)
#2 *Computing Surveys volume 10 #4* by Denning ($10 Assoc. for Computing Machinery)
#1 is a thorough introduction. #2 recommends a new graphics language.

Fantasy

#1 *The Colossal Computer Cartoon Book* by Ahl ($10 Creative Computing)
#2 *The Digital Villain* by Baer ($5 Addison-Wesley)
#1 is an anthology of cartoons about computers. #2 is a history of science fiction.

Artificial Intelligence

* #1 *Computers & Thought* by Feigenbaum & Feldman ($10 McGraw-Hill)
* #2 *Introduction to Artificial Intelligence* by Jackson ($20 Petrocelli/Charter)
#1 is easy; #2 is hard. #1 is old (1963); #2 is more up-to-date (1974).

Sorting

#1 *Computing Surveys volume 3 #4* by Dorn ($10 Assoc. for Computing Machinery)
#2 *Computing Surveys volume 11 #2* by Goldberg ($10 Assoc. for Computing Mach.)
#1 contains Martin's article on sorting. #2 contains Comer's article on trees.

How to organize a large data base

** #1 *Database* by Kroenke ($20 Science Research Associates)
#2 *Computer Data-base Organization* by Martin ($30 Prentice-Hall)
#1 compares the most popular software. #2 digs into fundamental techniques.

Numerical analysis

** #1 *Introduction to Numerical Methods* by Stark ($15 Macmillan)
* #2 *Elementary Numerical Analysis* by Conte & deBoor ($15 McGraw-Hill)
#1 is easier & explains round-off error. #2 gives advanced programming methods.

BOOKS THAT ELABORATE ON VOLUMES 7 & 8 (LANGUAGES)

FORTRAN

** #1 *FORTRAN 77 for Humans* by Page & Didday ($15 West)
*** #2 *The Elements of FORTRAN Style* by Kreitzberg & Shneiderman ($5 Harcourt)
#1 is fun, easy, & slow-paced. #2 contains advanced tricks to improve your style.

PASCAL

*** #1 *Introduction to PASCAL* by Zaks ($15 Sybex)
* #2 *Programming in PASCAL* by Grogono ($15 Addison-Wesley)
#1 is easier; #2 involves longer examples & more math. #1 includes UCSD PASCAL.

The easiest way to write programs in COBOL

** #1 *Introduction to COBOL (093-000181)* by Data General ($5 Data General)
*** #2 *Programming Standard COBOL* by Chai & Chai ($10 Academic)
#1 is short & sweet. #2 is more detailed.

The structured way to write programs in COBOL (explained briefly)

*** #1 *A Simplified Guide to Structured COBOL Programming* by McCracken ($10 Wiley)
** #2 *COBOL with Style* by Chmura & Ledgard ($5 Hayden)
#1 covers all the major facets of COBOL. #2 is a list ways to improve your style.

The structured way to write programs in COBOL (explained in depth)

** #1 *Structured ANS COBOL Part 1* by Murach & Noll ($20 Mike Murach)
** #2 *Structured ANS COBOL Part 2* by Murach & Noll ($20 Mike Murach)
Read #1. Then read #2, which continues where #1 left off.

PL/I

*** #1 *PL/I Programming with PL/C* by Davidson ($10 Houghton Mifflin)
* #2 *PL/I Structured Programming* by Hughes ($20 Wiley)
#1 is easier. #2 covers more topics.

Programming style, using PL/I with other languages

#1 *Programming Proverbs* by Ledgard ($5 Hayden)
** #2 *The Elements of Programming Style* by Kernighan & Plaugher ($5 McGraw-Hill)
#1 is a list of rules. #2 is a nasty exposé of bad programs that appeared in textbooks.

Strange numerical languages

* #1 *APL An Interactive Approach* by Gilman & Rose ($15 Wiley)
* #2 *Statistical Package for the Social Sciences* by Nie et al ($15 McGraw-Hill)
#1 explains APL for the IBM 370 & the IBM 5100. #2 explains SPSS.

SNOBOL

* #1 *The SNOBOL4 Programming Language* by Griswold et al ($15 Prentice-Hall)
#2 *String & List Processing in SNOBOL4* by Griswold ($15 Prentice-Hall)
#1 is an introduction. #2 is advanced & assumes you know what's in #1.

LISP

* #1 *Artificial Intelligence* by Winston ($15 Addison-Wesley)
#2 *Let's Talk LISP* by Siklossy ($15 Prentice-Hall)
The second half of #1 explains LISP easily with applications. #2 is harder & deeper.

Survey of popular computer languages

* #1 *Introduction to Programming Languages* by Peterson ($15 Prentice-Hall)
#2 *Programming Languages* by Tucker ($15 McGraw-Hill)
#1 covers 8 languages. #2 is similar, but omits BASIC, APL, & LISP, & adds RPG.

History of computer languages

#1 *Programming Languages* by Sammet ($25 Prentice-Hall)
#2 *Sigplan Notices volume 13 #8* by Sammet et al ($20 Assoc. for Computing Mach.)
#1 surveys 120 languages. #2 explains how 13 popular languages were invented.

Assembly language for microcomputers containing a Z-80 CPU

* #1 *Z80 Assembly Language Programming* by Leventhal ($15 Osborne)
** #2 *TRS-80 Assembly Language Programming* by Barden ($5 Radio Shack)
#1 is thorough. #2 is an easier introduction, for Radio Shack's model 1.

Assembly language for microcomputers containing a 6502 CPU

* #1 *6502 Assembly Language Programming* by Leventhal ($15 Osborne)
* #2 *Apple Machine Language* by Inman & Inman ($15 Reston)
#1 is thorough. #2 explains easily how to use machine language on the Apple.

Assembly language for IBM computers

* #1 *Assembler Language Programming* by Tuggle ($15 Science Research Associates)
#2 *System/360-370 Assembly Language (OS)* by McQuillen ($20 Mike Murach)
#1 is easier to understand. #2 is more practical, since it emphasizes I/O commands.

Assembly language for PDP computers

** #1 *Computer Organization & Programming* by Gear ($20 McGraw-Hill)
#2 *Introduction to Decsystem-10 Assembler Language Prog'ing* by Singer ($10 Wiley)
#1 explains the PDP-11 (& IBM 370). #2 explains the PDP-10 (& PDP-20).

Systems programming, using assembly language

** #1 *Systems Programming* by Donovan ($25 McGraw-Hill)
* #2 *Operating Systems* by Madnick & Donovan ($25 McGraw-Hill)
Read #1 first. Then read #2, which elaborates on the last chapter of #1.

How to use the popular operating systems

#1 *The CP/M Handbook* by Zaks ($15 Sybex)
* #2 *System/370 Job Control Language* by Brown ($15 Wiley)
#1 helps you use many microcomputers. #2 helps you use IBM maxicomputers.

WHERE TO GET BOOKS

Look for the books in your local library, or order them through your local bookstore or local computer store, or order them directly from the publisher, or join a computer book club.

In New York City, the largest store for computer books is the McGraw-Hill Bookstore (1221 Avenue of the Americas, New York, NY 10020, phone 212-997-1221). In Los Angeles, the largest store for computer books is Opamp (1033 N. Sycamore Ave., Los Angeles, CA 90038, phone 213-464-4322). In Boston, the largest stores for computer books are the Harvard Coop (1400 Massachusetts Ave., Cambridge, MA 02138, phone 617-492-1000) and the Tech Coop (84 Massachusetts Ave., Cambridge, MA 02142, phone 617-491-4230).

The two largest computer book clubs are The Library of Computer & Information Sciences (Riverside, NJ 08075) and the Computer Professionals' Book Club (c/o McGraw-Hill Book Clubs, P.O. Box 582, Hightstown, NJ 08520). They offer selected books at a slight discount, but you must pay a shipping charge, which almost equals the discount; and they don't tell you that some of the hardcover books are available more cheaply as paperbacks. As a come-on, they give you the first few books almost free; because that come-on is so good, I recommend that you join.

Here's where to find the publishers:

Publisher	City
Academic Press	New York, NY
Addison-Wesley	Reading, MA
Anaheim Publishing	Fullerton, CA
Ass. for Comp. Mach.	New York, NY
Book Co.	Lawndale, CA
Byte Books	Peterborough, NH
Compusoft	San Diego, CA
Creative Computing	Morris Plains, NJ
Data General	Southboro, MA
Everest House	New York, NY
Harcourt Brace Jov.	New York, NY
Hayden	Rochelle Park, NJ
Houghton Mifflin	Boston, MA
Hugo's Book Service	Chicago, IL
Macmillan	New York, NY
McGraw-Hill	New York, NY
Mike Murach & Assoc.	Fresno, CA
Osborne	Berkeley, CA
Petrocelli/Charter	New York, NY
Praeger	New York, NY
Prentice-Hall	Englewood Cliffs, NJ
Radio Shack/Tandy	Fort Worth, TX
Reston	Reston, VA
Science Research Ass.	Chicago, IL
Sybex	Berkeley, CA
West	St. Paul, MN
Wiley	New York, NY

Periodicals

NEWSPAPERS

The best newspapers about computers are Computerworld and Infoworld. Computerworld covers maxicomputers and minicomputers; Infoworld covers microcomputers. Computerworld is fat (about 120 pages per issue) and boring; Infoworld is thin (about 60 pages per issue) and fascinating. Subscribe to both, so you'll know all that's happening in the computer industry.

Both newspapers are owned by a company called CW Communications. So CW Communications controls all information about computers.

Computerworld comes every week. A single issue costs $1.25; a year's subscription costs $36.

Infoworld comes every two weeks. A single issue costs $1.25; a year's subscription costs $18.

To subscribe, contact CW Communications (375 Cochituate Rd., Framingham, MA 01701, 617-879-0700, 800-343-6474).

MONTHLY MICRO MAGAZINES

The best monthly magazines about microcomputers are Creative Computing, Personal Computing, Kilobaud Microcomputing, and Byte.

Creative Computing and Personal Computing are both published in northern New Jersey, and are easy to read. Kilobaud Microcomputing and Byte are both published in the teensy-weensy town of Peterborough, New Hampshire, and are more difficult.

Creative Computing is the most fun. Personal Computing does the best job of tutoring. Kilobaud Microcomputing is the most serious. Byte contains the most pages.

I recommend that you subscribe to all four. If you can afford only one, begin with Creative Computing. If you can afford only two, try Creative Computing and Kilobaud Microcomputing (so you taste both New Jersey and New Hampshire).

Here's how to contact them:

Creative Computing, P.O. Box 789-M, Morristown, NJ 07960, 201-540-0445, 800-631-8112
Personal Computing, 50 Essex St., Rochelle Park, NJ 07662, 201-843-0550, 800-323-1717
Kilobaud Microcomputing, 80 Pine St., Peterborough, NH 03458, 603-924-3873
Byte, 70 Main St., Peterborough, NH 03458, 603-924-9281, 800-258-5485

For information about a specific microcomputer, try 80 Microcomputing or Compute. 80 Microcomputing analyzes computers by Radio Shack; Compute analyzes computers that contain a 6502 CPU (such as Apple, Commodore, Atari, and Ohio Scientific). Contact:

80 Microcomputing, 80 Pine St., Peterborough, NH 03458, 603-924-3873
Compute, P.O. Box 5406, Greensboro, NC 27403, 919-295-9809

Summer course

BEST COURSE

The best course is the one I run every summer. It turns you into a computer expert in seven weeks. Because we're a public-service organization, the course is almost free!

Every summer, we improve the course further. The following description applies to the summer of 1981.

METHOD

To turn you into a computer expert in seven weeks, the course uses many unusual devices. The most important is the tutoring: you get one-to-one tutoring around-the-clock (days, nights, and weekends) for seven weeks. You also get, throughout the seven weeks, unlimited use of computers of all sizes: you use a maxicomputer, a minicomputer, and *every* popular microcomputer ever invented; we have *all* the popular manufacturers. Our computers understand a dozen computer languages. The computers are available 24 hours, and so are the tutors.

The course lectures dig into *every* computer topic. You'll learn:

how to program in BASIC, FORTRAN, COBOL, PASCAL, assembler, and other languages
how to select computer equipment that's best for your own office, school, or home
how to handle tough applications, by using advanced programming techniques
how the insides of the computer work, in detail
how to manage a computer center and handle the hassles that computers cause

For each topic, you'll learn the general philosophy and also the technical details.

TRACK RECORD

We've been running this course for eight years; many of the top professionals throughout the computer industry got their training from us. Our program, staff, and special texts have been praised by all the popular computer magazines.

Hard to believe, but true: by taking just the seven-week course, many students have obtained jobs as systems analysts, consultants, heads of computer sales departments, computer-center directors, and vice-presidents of computer companies, *even though they had little or no knowledge of computers or math before the course began*. That's because the course and its career counseling are so intense. We can't make any guarantees; but most students are pleasantly surprised.

HOUSING

The course meets in Connecticut, at Wesleyan University. If you're not from Connecticut— or if you're from the "wrong part" of Connecticut— we'll give you housing. Like the course, the housing is almost free. Again, a public service.

TIME

To participate in the program fully, you should arrive on June 29 and not leave until August 15. You can bring your family with you, and turn the experience into a family vacation.

If you can't spend that much time away from your current job or home, take one of our abridged versions. No matter which version you take, you'll find that you'll learn *more information* in a *shorter time* at a *lower price* than anywhere else.

YOUR PERSONALITY

You'll have lots of fun! Many of the older students say they feel "ten years younger" when they finish the course. That's because the course changes your personality as well as your intelligence. Our intense psychological methods bring out the best in you.

MANY VERSIONS

We offer many versions of the computer course. Choose the version that's best for *you*.

Our least expensive version: "526". Our least expensive version is called "526". It gives you seven weeks of one-to-one tutoring around-the-clock (days, nights, and weekends). The tutoring begins at 11 AM Monday June 29 and continues until 11 AM Saturday August 15. To make full use of the tutors, stay at Wesleyan full-time; but if you prefer, you can see the tutors less often. For example, if you're trying to juggle your family, a job, and the Wesleyan program simultaneously, and prefer to see your tutor just once a week, that's okay. The tutors are available continuously (24 hours per day); how much you use them is completely up to you. You also attend 18 hours of lectures. The lectures are scheduled 7:30-9 PM Monday and Thursday evenings. Since the lectures meet only during the evenings, your days are free— for tutoring or for your family. For all that instruction— using the computers, 47 days of around-the-clock tutoring, and 18 hours of formal lectures— the total tuition is just $100. That comes to just $2.13 per day. *When was the last time that somebody offered to tutor you around-the-clock for just $2.13?*

Our most thorough and fanciest version: "541-542". This is the version that I recommend the most highly. It's the same as "526", except that it includes 139½ hours of formal lectures (instead of 18) and costs $420 (instead of $100). The "formal lectures" include explanations, demonstrations, arguments, and guest-speaker interviews; they dig deeply into *all* aspects of the computer industry. Those 139½ hours of lecture propel your career much further ahead than the 18 hours of lecture in "526". Nevertheless, the price is still relatively low: since "541-542" costs $420 for 47 days, the price-per-day comes to just $8.94. If your family or job prevent you from attending the full course, come to the first half. The first half (which lasts about three weeks) is called "541"; the second half (which lasts about four weeks) is called "542"; each costs $210 (which is why the total cost for "541-542" is $420). So for just three weeks of intensive study, take just "541", which costs just $210. (If you can't come for even three weeks, come for the first *two* weeks and then quit. The first two weeks of "541" cover the most important topics.)

Our compromise version: "540". This version is a compromise between the previous two versions ("526" and "541-542"). Take it if you can't afford "541-542" but nevertheless want something fancier than "526". It includes 46½ hours of lecture and costs $310.

Our extremely advanced version: "550". If you think you're extremely advanced (because you've already been programming for many years, or because you've already spent a summer with us), take this version, which includes 46½ hours of advanced lectures and costs just $310.

COMPARISON CHARTS

This chart compares the different versions:

Version	Description	Prerequisite	Tutoring	Lectures	Tuition
526	our least expensive version	none	47 days	18 hours	$100
540	our compromise version	none	47 days	46½ hours	$310
541-542	our fanciest version	none	47 days	139½ hours	$420
541	the first half of 541-542	none	20 days	58½ hours	$210
542	the second half of 541-542	541 (or 540)	27 days	81 hours	$210
550	advanced version	542 (or 541 or 540)	47 days	46½ hours	$310

Here are the details:

Version	Tutoring begins 11 AM this day	Tutoring end 11 AM this day	Lecture days	Lecture schedule
526	Monday June 29	Sat August 15	July 2 - Aug 10	Mondays&Thursdays 7:30-9PM
540	Monday June 29	Sat August 15	July 1 - Aug 12	weekdays 11:45-1:15PM
541-542	Monday June 29	Sat August 15	July 1 - Aug 12	weekdays 10-11:30AM & 2-5PM
541	Monday June 29	Sun July 19	July 1 - July 17	weekdays 10-11:30AM & 2-5PM
542	Sunday July 19	Sat August 15	July 20 - Aug 12	weekdays 10-11:30AM & 2-5PM
550	Monday June 29	Sat August 15	July 1 - Aug 12	weekdays 9:15-10:45AM

OFFICIAL DESCRIPTION

Wesleyan's official blue catalog describes the computer program as follows....

Special program: WESLEYAN SUMMER COMPUTER INSTITUTE. Elementary and advanced courses, full-time and part-time, dig into *all* facets of computers intensively. Each student receives one-to-one tutoring, hands-on experience using computers of all sizes, candid advice from visiting specialists, and career counseling. The tutors are available 24 hours each day, to help each student at each moment. Because of the Institute's growing reputation for giving each student personal attention and in-depth instruction at low tuition, courses are certain to be oversubscribed, and applicants are therefore advised to apply early.

Courses in COMPUTER SCIENCE. Wesleyan offers unusual computer courses, through the Wesleyan Summer Computer Institute. The three introductory courses (526, 540, and 541) are similar to each other; the student can choose whichever best fits the rest of the student's schedule. 541 requires a full-time commitment; 540 is half-time; and 526 is a very light course, which meets in the evening for students who have other full-time commitments. After completing an introductory course, the student can progress to an advanced course: 542 or 550.

Each introductory course starts at the very beginning, and requires *no* background in computers or mathematics. The pace is quick. On the first day, the student is already programming the computer. Halfway through the course, the student chooses a topic on which to concentrate and works toward becoming an expert at it. Toward this end, the course includes many unusual features, such as around-the-clock one-to-one tutoring, a special accelerated text, seminars that meet at all hours of the day, guest speakers from all branches of the computer industry, a library of the best computer books and magazines, lectures staged as debates, and oral exams that refine the student's ability to speak publicly about computers. After the course ends, the student may use a hot-line for follow-up help and career questions. The course covers all four computer facets critically: how to communicate with a computer, by using a computer language (BASIC versus FORTRAN vs. COBOL vs. PASCAL); computer equipment (IBM's maxicomputers vs. DEC's minicomputers vs. Radio Shack's microcomputers); applications (to business, science, the social sciences, humanities, and arts); and management (how to buy a computer and manage a computer center in a business, in a school, or even in a home). In each area a wide range of problems is discussed (examples: Microsoft BASIC vs. BASIC-PLUS-2; FORTRAN IV vs. FORTRAN 77; COBOL 68 vs. COBOL 74; PL/I vs. PASCAL; Radio Shack vs. Apple vs. Commodore vs. Atari vs. other microcomputers; alphabetical lists vs. trees vs. hashing; structured programming vs. laissez-faire). Students have unlimited use of Wesleyan's computer center (which includes PDP-11 and PDP-20 computers, keypunches, teletypewriters, cathode-ray terminals, a graphics terminal, line printer, sound studio, and a dozen computer languages) plus several microcomputer laboratories (which include equipment of Radio Shack, Apple, Commodore, Atari, and many other microcomputer manufacturers, plus disks, printers, voice and music synthesizers, and a wide variety of the newest exotic devices). The course culminates in an individual project related to the student's personal interests. The advanced courses dig into the same topics again, but even more deeply.

To study all topics in detail, the student should take 541 and 542 the first summer, and 550 the second summer. The other courses (526 and 550) are less intensive.

Computer Science 526: INTRODUCTION TO COMPUTERS. This quick pleasant introduction is for students short of time.

Computer Science 540: COMPUTERS IN DEPTH. This is mainly for students who want a "solid" computer course but wish to take non-computer courses also.

Computer Science 541: ANALYSIS OF COMPUTERS AND PROGRAMMING. This course requires a full-time commitment for three weeks. It prepares the student to take 542, and is intended for students who are— or who wish to become— computer professionals.

Computer Science 542: ADVANCED COMPUTER STRATEGIES. This is a continuation of 541. Prerequisite: 541 or permission of the Computer Institute's director.

Computer Science 550: ADVANCED COMPUTER SEMINAR. This follow-up is mainly for students who enjoyed the previous summer and wish to return. Prerequisite: 542 or permission of the Computer Institute's director. (In 1981, this course is also open to any student who took "540" or "541".)

TEXTS

In addition to The Secret Guide to Computers, you'll use many other texts. If you don't want to pay for those extra texts, you can borrow them from my huge personal library. It contains hundreds of texts plus magazines, newspapers, newsletters, and private memos about computers. I've worked on that library for many years, to make sure it contains the best computer materials ever published.

CONSULTANTS

You'll be taught by a large team of consultants. No matter what topic you're interested in, you'll find a Wesleyan consultant who's an expert at it.

MANAGEMENT

How do we manage to cram so much into seven weeks? The answer: careful management. The lectures carefully avoid duplicating the texts: the lectures assume you've read the texts already, and therefore dig deeper. While some students are receiving personal tutoring, others are reading, others are trying to write programs, others are removing errors from programs they wrote already, others are eating dinner, and others are attending a seminar. We consultants stagger our own schedules— we eat and sleep at different times from each other— so that you can find a free consultant around-the-clock.

GUEST SPEAKERS

We have many famous guest speakers. Each speaker devotes many hours to Wesleyan. The speaker gives advanced technical information about computers, a candid commentary on the computer industry, and career guidance. We interview the speaker and ask many pointed questions. You join the speaker for dinner. The speaker meets with the students in small groups and then one-to-one; you chat with the speaker privately about your own personal needs are career plans.

CHOOSING HOUSING

Probably, one of your friends or relatives lives in Connecticut. If so, live with your friend or relative, and commute.

Or, if you prefer, live in Wesleyan's own pleasant housing. Several kinds are available. For most students, I recommend the "Butterfield College" housing. To live there through the entire 47-day program, you pay a total of just $175. That comes to just $3.72 per day. *When was the last time you paid just $3.72 per day for housing?*

If you're taking just half of 541-542, you'll be charged a total of just $90 for the three or four weeks.

PAYMENT

In addition to tuition, Wesleyan also charges a "registration fee", which is $25 per summer. (If you take two courses during the same summer, the total registration fee is still just $25.)

To enroll in the course, the only payment required immediately is the registration fee: only $25. By paying the $25, you reserve yourself a place in the course and also reserve yourself housing. Payment for tuition and housing isn't due until you arrive at Wesleyan.

REBATES

If you already own a computer (or buy one soon or borrow one) and bring it with you, we give you a rebate. The rebate is $30 per complete system, plus 2% of the list price (including the hardware, software, and even the television!), plus $30 per additional terminal (if any). You also get free insurance that covers all losses and repairs.

CREDITS

If you complete the course successfully, Wesleyan University will give you credits, which you can apply toward a master's degree.

FULL DETAILS

For full details, get the 16-page orange "Institute Brochure", by phoning me at 617-266-8128. Or use the coupon on the back page.

Blitz course

SUPER-INTENSIVE WEEKEND

If you can't take our summer course, take our "blitz" course. It condenses everything important about computers into *one super-intensive weekend!*

We offer the blitz course in Boston, New York, and other cities. We'll even come to your own home town— if you get together with your friends and co-sponsor it.

We give the course many times each year. Each time, we improve it further. The following description applies to most of 1981.

WHY?

The course covers *everything* important about computers, in one intensive weekend! Because we're a public-service organization, the course is almost free. If you're willing to spend one solid weekend learning about computers, we'll stuff so much into your head that at the end of the weekend you'll be an expert.

The course requires no previous knowledge of computers: we start at the very beginning, but move quickly, by using intensive psychological techniques.

We start with the two most popular topics: how to program in BASIC, and how to save money when you buy computer equipment. Then we dig into the many subtle problems of dealing with computers and the computer industry. Throughout the course, you gain candid insights not available anywhere else.

TRACK RECORD

We've offered this successful course many times, and it's significantly changed the careers of its students. Example: a beginner who knew nothing about computers learned enough during the single weekend so that he immediately got a job as marketing director of a large computer company. I can't guarantee the same will happen to you; but I *do* guarantee that you'll learn more about computers from this weekend than from any other two-day experience you'll have anytime anywhere. It will significantly affect your career.

HOURS

We meet ten hours on Saturday (from 9:15 AM to 7:15 PM), and the same hours on Sunday. So the entire weekend course gives you 20 hours of instruction.

You learn while you eat. Bring sandwiches with you, since the instruction will continue straight through the lunch and dinner hours.

STAFF & EQUIPMENT

You'll be taught by me and my consultants. You'll put your hands on all the popular microcomputers (Radio Shack, Apple, Commodore, Atari, and others), use fancy peripherals (tapes, disks, printers, and voice and music synthesizers), and hear salesmen and consultants argue pro and con about each manufacturer. We'll analyze their arguments. We'll also help you advance your computer career.

YOUR OWN COMPUTER

If you're a lucky gal or fellow, who owns a computer already, bring it with you. Bring your programs also. You'll get personal criticisms of your programs. You can swap programs with other computerists.

EXACT PRICE

The entire weekend course costs just $58. You get 20 hours of training. So you're paying just $2.90 per hour. No other organization offers an extensive computer course at such a low price.

If you don't want to spend $58, you can spend just $29, which gets you all of Saturday. (On Saturday, I cover all the essentials; on Sunday, we dig deeper.)

METHOD

You learn by the "blitz" method: a rapid-fire lecture-demonstration-question-argument-and-answer method, that begins by assuming you know absolutely nothing about computers and teaches you so quickly that, by the end of the first half-hour, you already know how to make the computer imitate your own personality!

FORMAT

The course consists of fourteen sessions:

Time	Saturday	Sunday
9:15-10:45	Beginning programming	Data files
11:15-12:45	Equipment	Business packages
1:15- 2:15	Intermediate programming	Advanced programming
2:45- 3:30	Popular microcomputer companies	Communications and I/O
4:00- 4:45	Subtle programming	Popular languages
5:15- 6:00	Alternative computers	Alternative languages
6:30- 7:15	Fun applications	Careers

Each session includes lecture, discussion, demonstrations, and arguments. Between sessions, you put your hands on the computers, browse through books and magazines, get tutored, and receive personal advice from the consultants.

CONTENTS OF EACH SESSION

Beginning programming: program your personality, printers, tapes, disks
Equipment: CPU, memory, I/O devices, how much memory do you need?
Intermediate programming: DATA, FOR, vagueness, functions, randomness, color
Popular microcomputer companies: Radio Shack, Apple, Commodore, Atari
Subtle programming: subscripts, subroutines, style, how to tell a "good" program
Alternative computers: other microcomputers, minicomputers, maxicomputers
Fun applications: games, graphics, therapy, music, speech, networks, schools
Data files: sequential, random, alphabetizing, trees, hashing, data bases
Business packages: PFS, Visicalc, word processing, accounting, consultants
Advanced programming: dialects, INKEY$, round-off, ASCII, formats, POKE
Communications and I/O: printers, terminals, modems, interfaces, baud, duplex
Popular languages: BASIC, FORTRAN, COBOL, PASCAL, interpreters, compilers
Alternative languages: high-level, machine, octal, hexadecimal, assembly
Careers: kinds of jobs, tricks to get them, schools, what to read, clubs, help

TEXTS

To get the most out of the course, read the first four volumes of The Secret Guide to Computers. They give you enough background so that you can ask intelligent questions during your 20-hour blitzing.

TAPES

You can bring a tape recorder and tape everything we say. But why bother? If you ever forget any of the info, call us on the phone, free, and we'll review it. Yes, you can even tape the phone calls.

REFUNDS

If you pay for the course and don't show up, I'll mail you a 100% refund. All refunds are mailed *immediately*.

FULL DETAILS

For full details, phone me at 617-266-8128, or use the coupon on the back page.

Old version

PROGRESS

A new version of volume 1 came off the press in May 1981. A new version of volume 2 came off the press in June 1981. If you bought volumes 1 and 2 before then, you have the older version, which contained a few errors. Make these corrections....

VOLUME 1

On page 33, the program's line 170 mentioned F$; it should have mentioned D$ instead. On page 75, paragraph 5 said, "Eliminating line 110"; it should have said, "Eliminating line 210". For a list of other errors, read the "Apology" on page 6 of volume 2.

VOLUME 2

The bottom of page 36 should have said:

> So if T has passed Saturday (which is 50), we want T to become 2 (for Sunday); and if there are more days left in the month, we want the computer to press the return key (to start a new week):
>
> ```
> 72          IF T>50 THEN T=2: IF D<L THEN PRINT
> ```

So change line 72. Change it also at the top of page 37. At the bottom of page 37, change line 72 to this:

```
72          IF T>50 THEN T=2: IF D<L THEN PRINT: PRINT: PRINT
```

Page 41 said that in "mode 23" the X coordinate "can go from 0 to 79". It should have said instead that the X coordinate "can go from 0 to 159".

The explanation of Atari graphics was unnecessarily complicated. For a better explanation, get the new version of volume 2.

On page 73, the last paragraph (which said "An explanation...") should be replaced by this text:

answers GEORGE WASHINGTON or simply WASHINGTON or G. WASHINGTON or GENERAL GEORGE WASHINGTON or PRESIDENT WASHINGTON or MARTHA's HUSBAND? Make the instructions clearer:

```
WHO WAS THE FIRST PRESIDENT OF THE UNITED STATES?
GIVE JUST HIS LAST NAME.
```

The user is trying to be funny or sabotage the computer. Instead of inputting his name, he types an obscene comment. When asked how many brothers and sisters he has, he says 275.

It's your responsibility as a programmer to make sure that the directions for using your program are clear, and that the program rejects ridiculous input. For example, if your program is supposed to print weekly paychecks, it should refuse to print checks for more than $10000. Your program should contain these lines:

```
20 INPUT X
30 IF X>10000 THEN 100
...
100 PRINT X"IS QUITE A BIG PAYCHECK!  I DON'T BELIEVE YOU."
110 PRINT "PLEASE RETYPE YOUR REQUEST."
120 GO TO 20
```

Lines 30 and 100-120 are called an *error-handling routine*. Your program should contain several, to prevent printing checks that are too small (2¢?) or negative or otherwise ridiculous ($200.73145?).

To see how your program reacts to input that's either garbage or unusual, ask the person sitting next to you to run your program. He might input something you never thought of.

Coupon

SHARE

Share your secret: photocopy this page, and give it to a friend. If you know a young (or young-at-heart) computerist who's having a birthday, Christmas party, or graduation, turn these volumes into a present: send me the computerist's name, address, and either a greeting card or what you want me to say.

VOLUMES

This tenth edition of *The Secret Guide to Computers* consists of eight volumes:

Volume	Topic
1: Popular BASIC	the easiest way to program, using many dialects of BASIC
2: Hassles in BASIC	advanced BASIC, data files, graphics, programming style
3: Popular Systems	kinds of computer equipment, how they work, buzzwords
4: Hassles in Systems	the newest computers, operating systems, your career
5: Popular Applications	what computers have done and can do
6: Hassles in Applications	advanced programming techniques for advanced applications
7: Popular Languages	FORTRAN, COBOL, PASCAL, comparison with BASIC
8: Hassles in Languages	assembly languages, dozens of high-level languages

Each volume costs $3.70. The complete 8-volume set costs $29.60.

DISCOUNTS

To save money, get together with a friend:

20% discount if you order 2 complete sets (or any 16 booklets)
30% discount if you order 4 complete sets (or any 32 booklets)
40% discount if you order 10 complete sets (or any 80 booklets)
50% discount if you order 50 complete sets (or any 400 booklets)
60% discount if you order 500 complete sets (or any 4000 booklets)

POLICIES

Do *not* send extra money for postage and handling: I'll pay for them myself, even if you live in a foreign country. If you're not sure whether to order a book, go ahead: you can return unused books at any time, for a 100% refund.

CUT OUT THIS COUPON!

Circle the volumes you want: 1 2 3 4 5 6 7 8 $3.70 each; $29.60 for all eight. Mass. residents add 5% sales tax.

FREE brochures are available on the following topics; check off the ones you want:

___ the summer course

___ the blitz course in Boston

___ the blitz course in New York City

___ the blitz course in other cities

___ later editions of *The Secret Guide to Computers*

___ *The Secret Guide to Tricky Living* (Volume 1: "Sex". Volume 2: "Everything But".)

On the other side, write your name, address, and any comments you have (pro or con) about *The Secret Guide to Computers*. Make check or money order payable to Russell M. Walter. Send to Russ Walter, 92 Saint Botolph St., Boston, MA 02116.

WELCOME...

to the NEW AGE...

of PLEASURABLE COMPUTING!

If you've reached the end of this book,
you've reached the end of your problems!